CENTERLINE MAINTENANCE PROCESS

Dec. 2001
1st Rev. Dec. 2004
2nd Rev. Nov. 2007

Mark Bradley
GIS Analyst
City of Indianapolis/Marion County

FORWARD

This document was originally written in 2001 in response to a request from my supervisor to document the processes I use to maintain our street centerlines. Much has changed since that initial version was written. Back then, we were using ESRI coverages to store our GIS data. In 2004 I updated this document to describe the changes that were made when we moved to the geodatabase model. This second revision to the document attempts to describe the current state of the centerline maintenance process. Only minor changes have been made to the process since the last revision. Besides describing those changes, this version expands on a few topics, in an attempt to explain them more fully, and includes more cross-references.

This document is not meant to be a "how-to" manual or model for other cities, just an explanation of how things are done here. I have tried to cover all facets of the centerline maintenance process in great detail, and it is my hope that this document will serve as a means for someone else to be able to step in and assume responsibility for the centerlines, should I no longer be in this capacity.

I have been an employee for the City of Indianapolis/Marion County for 29 years. The first 14 years of my tenure I was involved in inspecting new street construction; the remainder of this time I have been involved with documenting various aspects of streets, using computers. Because of my experience I guess, I have been given a lot of flexibility in how I carry out my responsibility of maintaining our street centerlines. Consequently, you will see some references in this document as to how "I" do something a certain way. A lot of my procedures and policies I have developed on my own; hopefully and humbly with the City of Indianapolis' best interests in mind. So, some of the stuff in here is my opinion alone, and I don't claim to be an expert.

My expertise is not in engineering or real estate. Consequently, I may not always use industry-standard terms within this document for items relating to those professions.

Being currently an employee of the GIS Division of the Information Services Agency of the City of Indianapolis/Marion County, I have made references in this document to persons and organizations with which the other employees of the GIS Division would be familiar.

TABLE OF CONTENTS

INTRODUCTION

Like Diogenes, I search for solutions to geospatial problems I face,
Wasn't ESRI software developed primarily to help us analyze space?
Yet managing data and preparing maps is the labor of far too many and not just a few,
Is maintaining cadastral data and mapping street centerlines all some users can think of for their GIS to do?

Jay Morgan, 2007

The street centerlines for the City of Indianapolis were originally created in 1986, when the Indianapolis Mapping and Geographic Infrastructure System (IMAGIS) was formed. (IMAGIS is a multi-participant, public-private Geographic Information System (GIS) consortium for Indianapolis and Marion County, Indiana.) Those of us working in City government in Indianapolis commonly refer to the street centerlines as simply "the centerlines," even though other GIS layers (e.g., sewers) are also centerlines. Since the street centerlines are considered to be the most important of our layers (in my humble opinion, anyway!), they are awarded the honor of being referred to as simply "the centerlines." Thus, in the remainder of this document, I will use the word *centerlines* to refer to street centerlines.

In 1997 I took over exclusive maintenance of our centerlines and, in fact, that is my primary responsibility and consumes the bulk of my time. Indianapolis is fortunate to be able to allocate a full-time person to this task, and as such, I have had a lot of time to develop and refine our centerlines. We have come a long way in 20 years.

The current centerline model has been developed to fulfill several purposes. These include:

- Cartography — The ability to produce geometrically accurate and aesthetically pleasing maps.
- Analysis — The ability to derive information using the centerline attributes.
- Geocoding — The ability to locate street addresses spatially by letting the system search the address-related attributes.
- Routing — The ability to find the most desirable path between two or more points, taking into account various impedances.
- Historical Research — The ability to know where streets used to exist in the past, that no longer exist.

In order to fulfill these purposes, several goals for the centerlines have been established by myself and others, and are simultaneously being worked on or have been completed. Refer to the section on goals at the end of this document to see a list of these.

Data Structure

Many years ago the City of Indianapolis standardized on ESRI software to fulfill its GIS needs. Layers were originally maintained as shapefiles, but for the most part have been transferred to geodatabases. Mission-critical layers have been moved to ArcSDE geodatabases running on ORACLE, and editing is done in ArcMap. These layers are allocated their own server, named *IMCOSL04*, reserved for layers that are updated daily. This server is intended for editing only; users are not allowed access to it. I access the centerlines on this server by way of a database connection called *sdeDynamic*.

Periodically, the layers on the *IMCOSL04* server are copied to another server, *IMCOSL03*, which provides read-only access to our users. This arrangement provides enhanced data access and security. These copies of the original layers are accessed through a database connection called *sdeStatic*. Also on the *IMCOSL03* server are all the other layers we maintain; that is, those that are updated infrequently (e.g.,

annually).

All our layers in the ArcSDE geodatabase format reside within ESRI feature datasets. These datasets are named using the ORACLE convention of *Owner.DatasetName*. The owner (account name) we use is "CCGIS" (for "City/County GIS"). Our street centerlines reside in a feature dataset called *CCGIS.IndyStreets*, visible when I connect using *sdeDynamic*. (There are several other feature datasets accessed by way of that connection, including *CCGIS.IndyParcels* and *CCGIS.Sewers*.)

The centerlines themselves are a feature class within the *CCGIS.IndyStreets* dataset, and they are named *CCGIS.fcStreets* (where again, "CCGIS" is the owner name, "fc" stands for "feature class," and "Streets" is the layer name). Unless otherwise noted, all the information in this document pertains to this layer. (For a map of this layer, see Figure 75 later in this document.) The cluster tolerance for the layer is set at 0.01 feet.

Also in the same dataset are several other associated layers (features classes) that interact with and are edited in conjunction with the centerlines. The names of these layers also start with the owner name "CCGIS." Some of the layers contain an "fc" prefix in their name, and some don't. (There is no particular reason for this — it is simply how our database administrator, Chuck Carufel, named them when they were created.) These associated layers are:

fcStreetsOOC

fcIntersection

fcCulDeSac

fcCulDeSacOOC

Seclines

SurveySectionCorners

PreliminaryAlleys

PreliminaryAlleysToo

fcTrafficCounts

fcStreetClosures

Trails

IndyStreets_Topology

Throughout the remainder of this document, I will generally omit the owner name and the "fc" prefix when referring to these layers. The *StreetsOOC* layer is explained in the section entitled *Geographic Extent* on page 4 in the next chapter. The rest of the layers except *IndyStreets_Topology* are explained in the section *Associated Layers* beginning on page 71. Topology is explained on page 96.

The Master Address Database

The City of Indianapolis has implemented a "Master Address Database" (MAD), running on ORACLE, which is intended to be a comprehensive collection of all valid parcel, building, and street addresses (address ranges) for which the City/County municipality has jurisdiction. (Besides Marion County, this includes certain properties slightly outside the county for which Indianapolis provides sewer service.) As new

applications are designed for the enterprise, if they contain an address component, they must hit against this database, to verify that any addresses used are valid. The MAD is designed to include other types of addresses as well, such as incident addresses, Work Order addresses, complaint addresses, etc.

The parcel, building, and street address range addresses in this database come directly from their respective GIS layers. This is accomplished by way of some custom tools that we had a consultant design for us. Whenever parcel, building, or street addresses are edited within one of these GIS layers, one of these custom tools is employed, which automatically saves the changes to the layers and the MAD at the same time. (See *Populating Centerline Attributes*, page 133.) There are also triggers in the MAD that propagate these changes to other ORACLE databases; namely, the Tidemark permitting system and the Hansen Infrastructure Management System.

Since I am solely responsible for maintaining our centerlines, I must be careful that the address ranges include all existing parcel and building addresses, as far as I can determine. However, I must balance this need with the need to not make the address ranges unnecessarily large, where they would contain addresses beyond the range of those that actually exist.

Brian Schneider is responsible for maintaining our parcels layer. He (and others) are also responsible for assigning addresses to new buildings and parcels. The way the Master Address Database is designed, Brian is prohibited from entering these addresses if they do not already exist on the centerlines. This is intended to be a safeguard policy. Consequently, there is a process that must be followed when new streets (and addresses) need to be added to the database. This is explained in the *New Streets* section on page 101.

A Note About the Illustrations

I have discovered through experience that some of the cursors for the tools I use in ArcMap are white, and others are black. This means if you use a white or black background in your ArcMap data frame, some cursors will be invisible. Therefore, I choose to use a gray background when editing to avoid this problem. Also, I feel that a gray background is easier on the eyes than a white or black background. However, for the illustrations in this document, I have employed a white background, so that if the reader wishes to print this document, the illustrations won't use up a lot of toner or black ink. A few illustrations that contain yellow lines are shown with a gray background, however, so that the yellow lines will show up better.

I have employed standard color symbolization in all the illustrations. Edge of pavement is depicted in green, and parcel boundaries in blue. Building footprints are shown in light gray. Street centerlines for most streets are black, with various other colors being used for certain types of streets. (See the section *Setting Up a Typical Editing Session* on page 103 for more on this.) All street centerlines are symbolized with arrowheads indicating the digitizing direction.

North is always at the top in the drawings.

BUSINESS RULES

The most important business rules pertaining to centerlines were developed during a series of meetings held between 1997 and 1999 by a group known as the City/County GIS Committee. (I have copies of the minutes on file.) I have added a few minor rules since then, and some of the original rules have evolved over time since they were originally created.

Geographic Extent

In 1970 the City of Indianapolis municipal government and Marion County (where Indianapolis is located) government were consolidated into one unified governing structure, deemed "Unigov." Basically, this extended the jurisdiction of all City services (with the notable exception of the Police Department) to the Marion County boundary. (No part of Indianapolis extends outside Marion County.) For this reason, the centerlines that are maintained by the City cover all of Marion County. This is the *Streets* layer.

As explained in the Master Address Database section of the introduction to this document, there are also a few centerlines in the *Streets* layer that lie outside Marion County. These centerlines are located in areas where there are sewers that are part of the City of Indianapolis' sewer network, that extend just beyond the boundary of the county.

We also find it desirable to maintain selected centerlines from the counties surrounding Marion County. These are contained in a separate layer called *StreetsOOC* ("Streets Out-of-County"). These centerlines are obtained using a variety of methods, using whatever means available. The layer started out as a collection of all the streets within a one-mile "collar" outside the Marion County boundary. These streets were digitized from our own aerial photos, which were specifically flown to include the collar area. To this layer were added all the centerlines in the surrounding counties that are a part of the Marion County *Official Thoroughfare Plan*. These were obtained from some files that a vendor digitized under contract to the Department of Metropolitan Development (DMD). They were digitized from scanned USGS topographic quadrangle maps, then reprojected into our Stateplane coordinate system and appended to our data. Because they were digitized from 1:24,000 USGS quad maps, they are somewhat rough, but they serve their purpose. Finally, from the centerlines we received from Hamilton County, our neighbor to the north, I also appended another mile of streets to the collar, so that our centerlines extend two miles into Hamilton County. This was done at the request of Layne Young on our staff, who had occasion to make many maps of that area for our Fire Department.

It is generally understood that the centerlines in the *StreetsOOC* layer are not as accurate or reliable as the centerlines for our own county, because they are outside our jurisdiction. Also, although this layer carries the same attributes as our *Streets* layer, many of those attributes are not populated in this layer. However, we are in the process of developing data-sharing agreements with the surrounding counties, from which we are obtaining various GIS layers. We also are receiving updates from TeleAtlas for all our surrounding counties, in exchange for providing them our centerlines. If and how all this data from the surrounding counties will be integrated with our data has not been explored yet, but at least it will be available. Perhaps if we eventually get good data from all the surrounding counties, we will dispense with the *StreetsOOC* layer entirely. It is not linked to any external databases, like the regular *Streets* layer is.

Inclusiveness

All streets and alleys will have a centerline. (For the purposes of centerlines, alleys are treated just like streets.) Whether or not a street is named doesn't matter. Also, whether or not a street is improved doesn't matter. It doesn't even matter if the street still exists or not. Centerlines are included for streets that are still in the planning stage (for public streets, those not yet platted), public platted streets (constructed or not), public vacated streets (constructed or not), and streets for which the pavement has been removed. Centerlines for newly-platted developments are entered as soon as their design is approved and they're made available to me. Centerlines of historical streets (removed) are added as time and resources permit.

Deleting Centerlines

Centerlines are normally never deleted, so that they may be used for historical reference. If a street's pavement and/or right-of-way no longer exists, this is denoted on the centerline by the *Operational Status* attribute value. (See page 41.) There are three situations where a centerline can be deleted:

- If a centerline is created by mistake.
- When streets are rebuilt, if the new alignment of the pavement deviates only slightly from the old alignment, the old alignment is deleted. See Figure 1, next page. However, if the street has right-of-way (is not a private street), and it is not an interstate or highway, the new pavement alignment must deviate significantly in order to justify changing the centerline alignment and deleting the old configuration.
- Once in a great while a new subdivision will be platted, but before the subdivision is constructed, a different subdivision (or a completely different configuration of the same subdivision) will be platted in the same location. In this situation, the original platted subdivision centerlines are deleted.

Geometry

Note: Throughout this document, I will use the term "node" to refer to the point where two or more centerlines meet. (This is not to be confused with "intersections," which have a different meaning as used in this document, and which are discussed later.) In the old ESRI coverage model days, an actual node feature existed at these locations; utilizing the current geodatabase model, nodes are no longer used, and they no longer exist as actual features at these locations.

Alignment

For streets without right-of-way, the centerline will follow the center of pavement. If a street contains turn lanes, passing blisters, etc. that temporarily increase the pavement cross section width, the centerline will follow the "theoretical" center of pavement, as if the extra pavement width did not exist.

For streets with right-of-way, the centerline generally will follow the center of right-of-way. If the right-of-way changes, the centerline will shift correspondingly, unless the deviation is just for a short distance, in which case the centerline will not be shifted. However, for thoroughfares that generally follow Public Land Survey System section lines, the centerline will coincide with the section line for its entire length. This is to avoid having to create many shifts in the centerline alignment, since thoroughfare rights-of-way are seldom uniform, and often change due to the addition of right-of-way for new adjacent subdivisions or other developments. See Figure 2, next page.

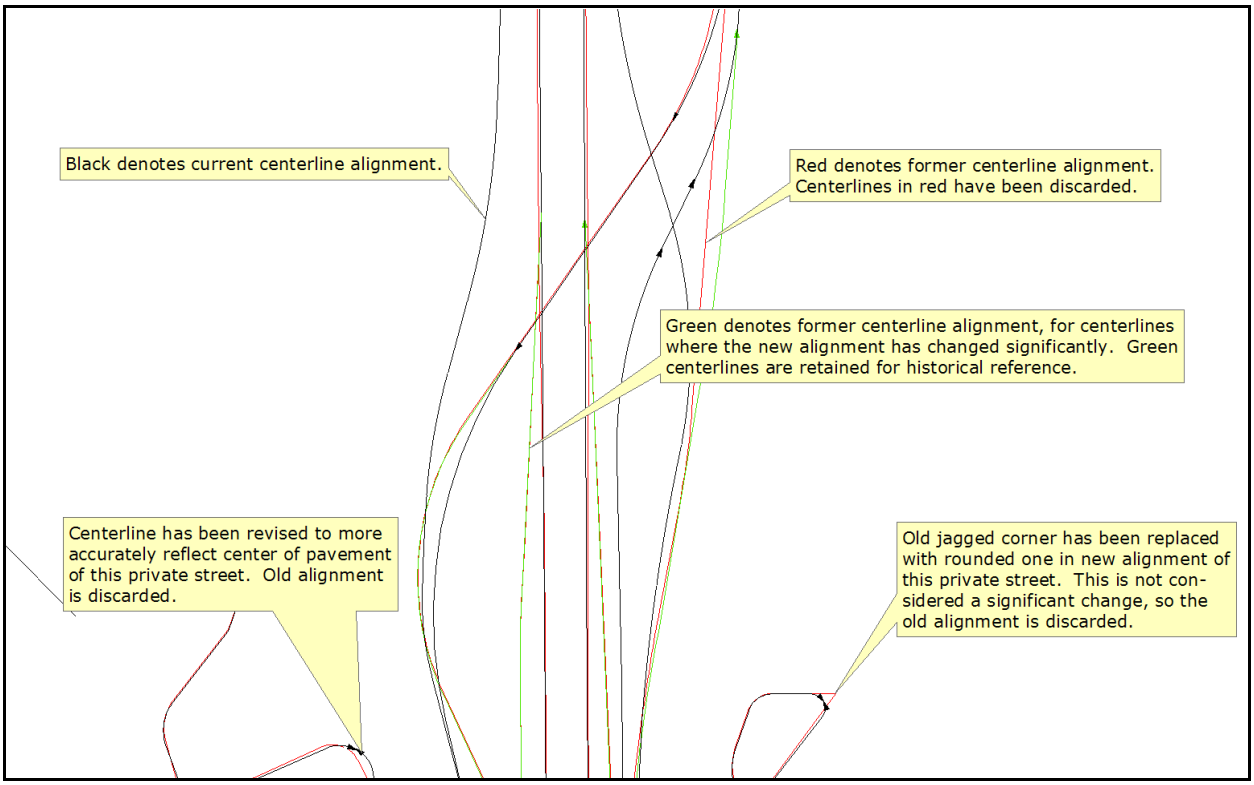


Figure 1 - Comparison of former centerline alignments from an old version of the centerlines (shown in red), and the current version (shown in black and green).

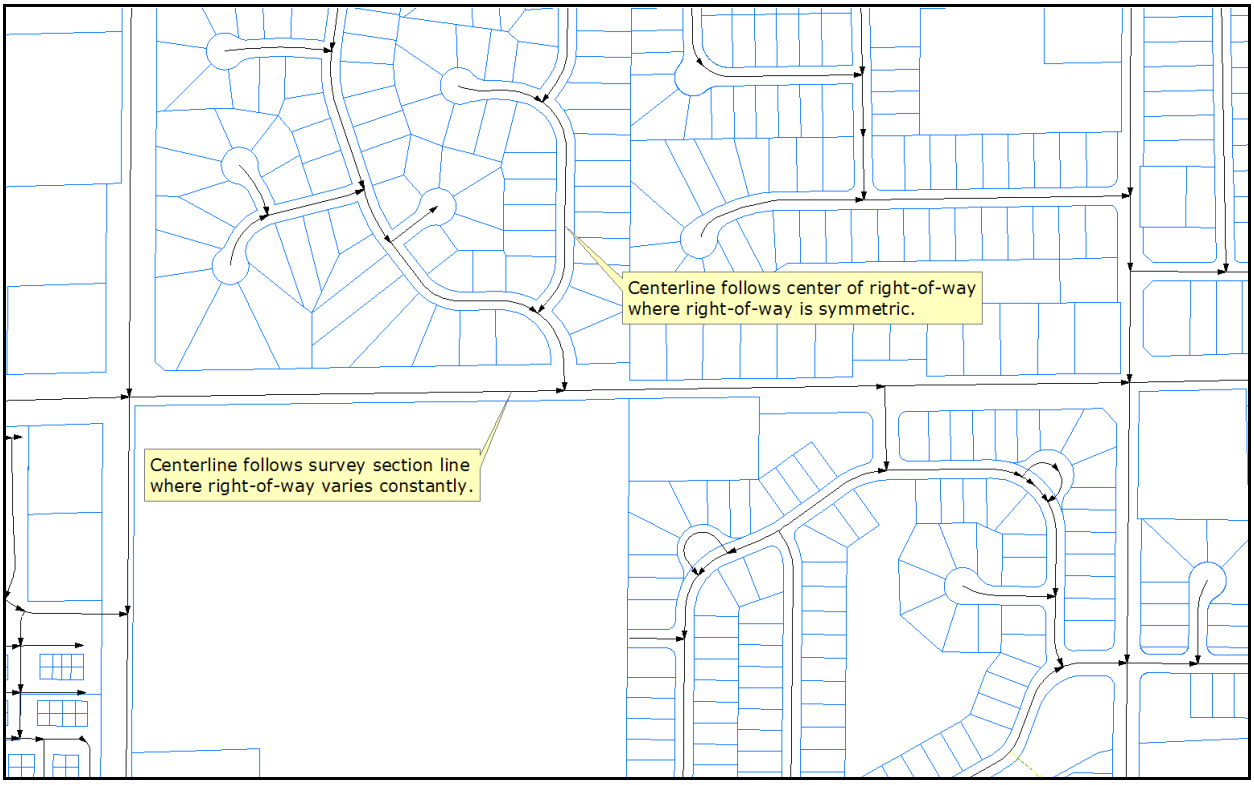


Figure 2 - Typical centerline alignments.

Centerlines without right-of-way are not allowed to change direction more than a few degrees without the use of a curve. There are three reasons for this:

- Appearance. I think rounding off corners makes more pleasing maps.
- Vehicles cannot turn on a dime, but require a certain amount of room to turn. I like the centerlines to reflect this.
- In the future, if we ever use our centerlines as the basis for certifying the mileage of our streets (possibly for gas tax allocation), we need the centerlines to match the actual length of the streets as closely as possible. Approximating curves with tangents can lead to under-reporting or over-reporting street length.

Figures 3 and 4 on the next page illustrate an unacceptable and an acceptable centerline configuration.

Figure 5 two pages over shows an example of an acceptable sharp turn in a centerline. This example is acceptable because this street has right-of-way that also contains a sharp turn, and following the center of right-of-way (when the right-of-way is consistent) governs over appearance.

If the right-of-way of a *built* street changes between intersections, and that change is the only change between intersections (or there is a long distance to the next change), the transition of the centerline from one right-of-way width to the other will consist of a symmetrical "S" curve. This is in keeping with the three reasons stated above. The center of the "S" curve will align with the exact point where the right-of-way changes width. See Figure 6 two pages over.

However, if the street is not built, then the transition will be depicted with right angle turns. Since the street is not built, it will usually not be included on maps; thus, the reasons noted above don't apply.

(Note: There are many existing centerlines in our database that currently violate this policy of not having sharp curves. That is because these are legacy centerlines that were created before this policy was implemented. Over time, it is my hope to revise as many of them as possible to conform with the policy.)

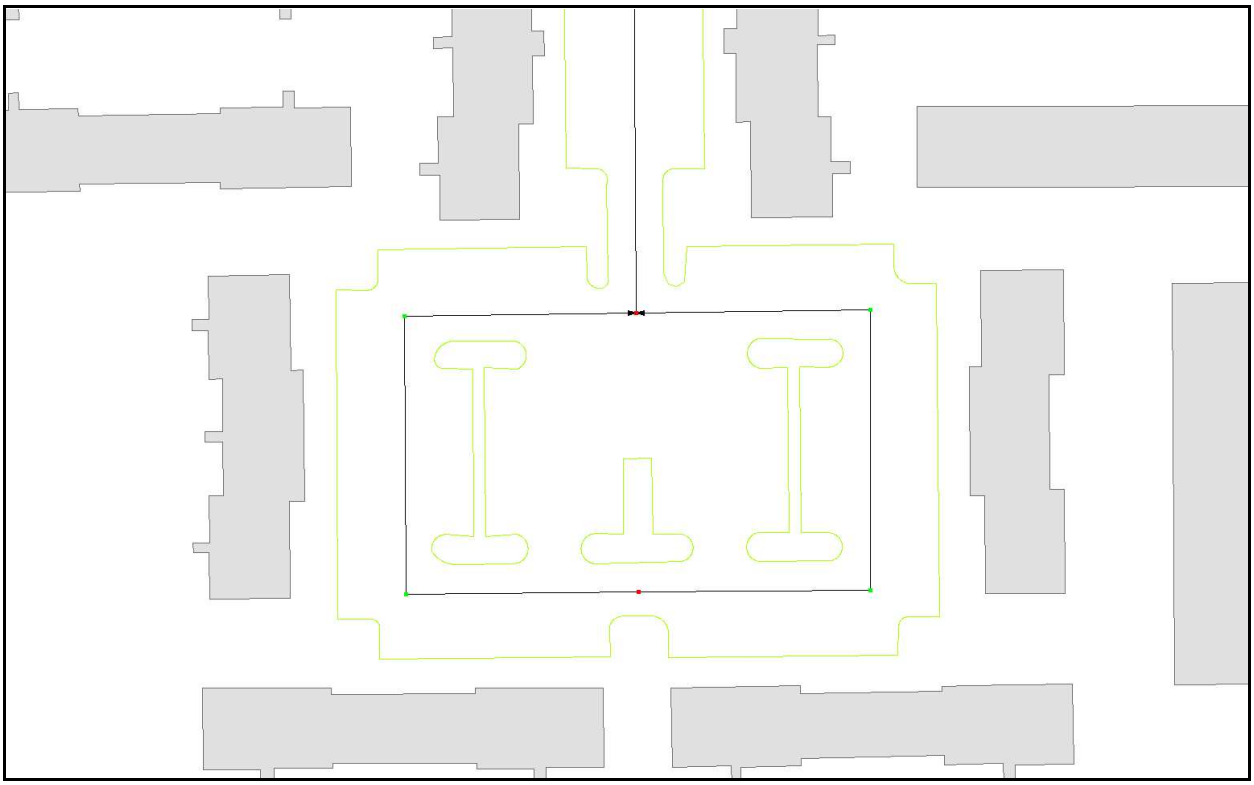


Figure 3 - Not this...

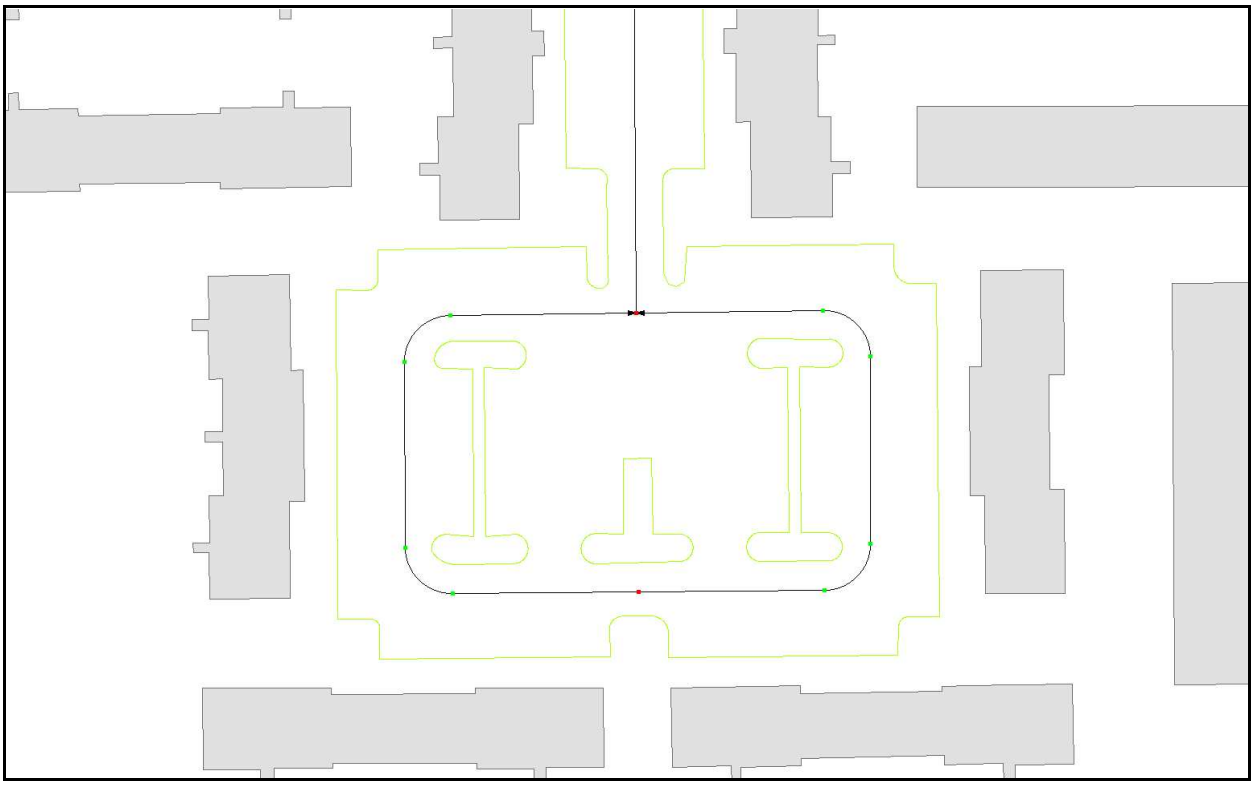


Figure 4 - ...but this!

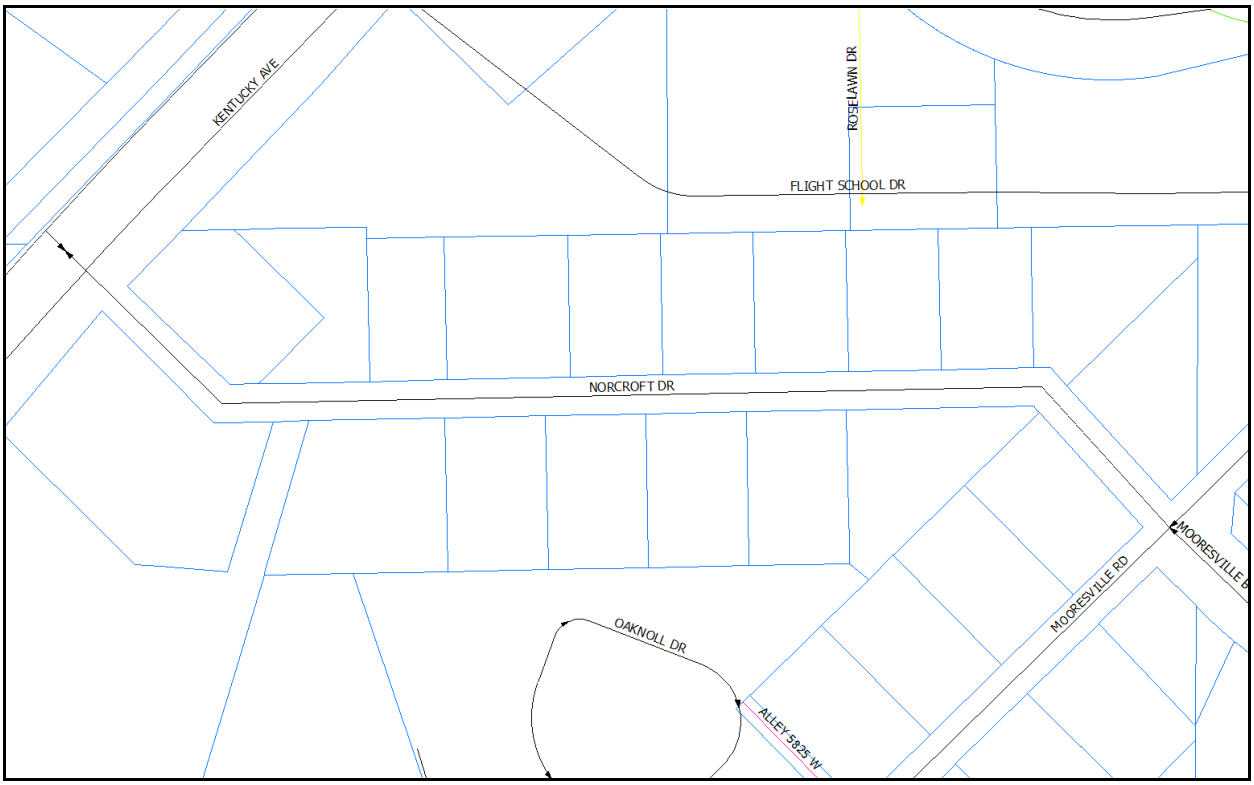


Figure 5 - Example of a street whose right-of-way contains sharp bends.

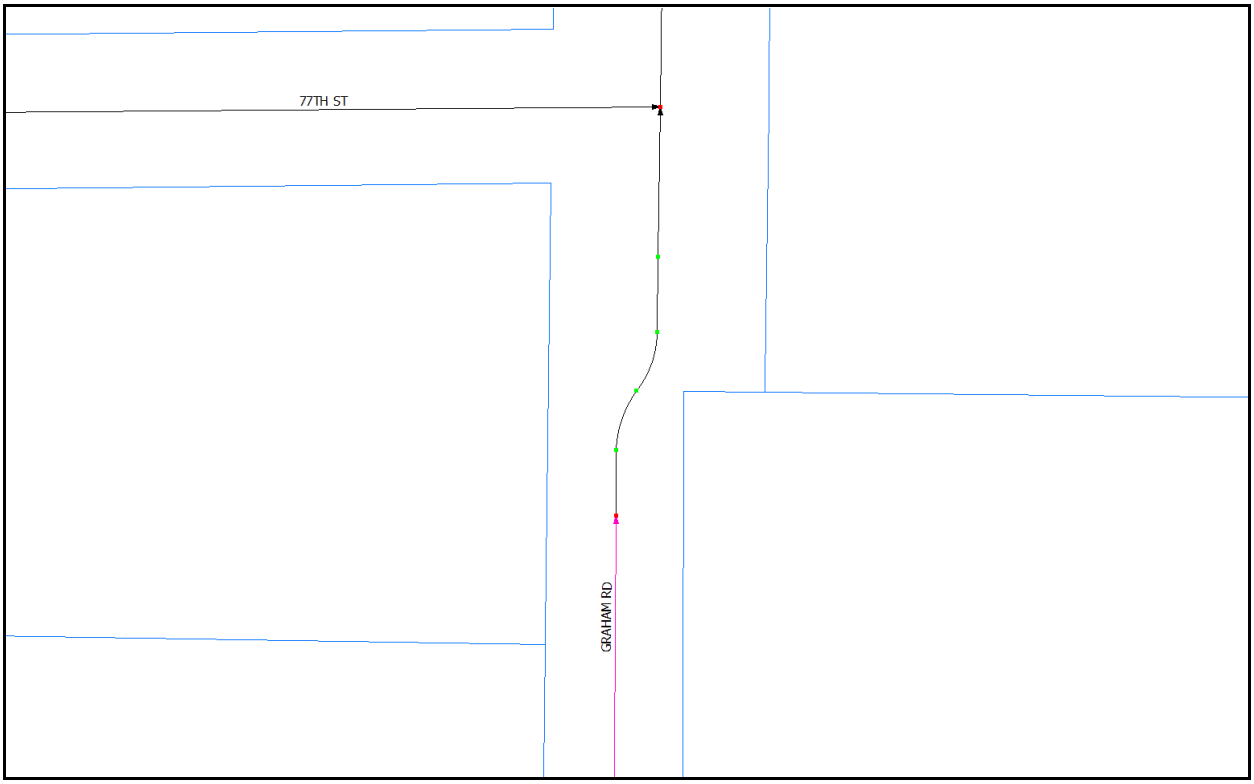


Figure 6 - A transition from one right-of-way width to another. Notice the "S" curve is centered on the point where the right-of-way width changes.

Multiple Centerlines

A major issue concerning street centerlines is what to do about streets that consist of multiple, parallel pieces of pavement (e.g., divided streets). Should each piece of pavement have its own centerline? I consider three factors when deciding this issue: routing, attributes, and cartography.

Concerning routing, I think in terms of traffic movement. If one of the purposes of your centerlines is to model traffic flow, you need to be able to represent restricted movements (one-way traffic and restricted turns). These criteria will sometimes dictate extra centerlines (and intersections), to be able to model this accurately.

Secondly, you may want to be able to assign different attribute values to the separate pieces of pavement of a street. The City of Indianapolis utilizes a Pavement Management System to help program capital street improvement projects. The program requires various kinds of information related to streets, stored in attributes, in order to function. Although this system is not currently linked to our centerlines, the use of multiple centerlines for separate pieces of pavement of a street will facilitate this linking in the future. As an example of how this might be useful, in the case where a resurfacing contract calls for resurfacing the main part of a street that is heavily traveled, while ignoring a secondary piece of less heavily traveled pavement of the same street (like a frontage road), we can map exactly which portions of the street were resurfaced. Also, the main pavement section is likely to have a different Pavement Condition Index (PCI) than the secondary one, which could be assigned as an attribute of each separate segment.

Thirdly, there may be situations where you want to use multiple centerlines to represent a street cartographically, just to better represent reality, where the first two conditions are not met.

Based on these three factors, my policy is: where a street consists of multiple parallel pieces of pavement, regardless of whether or not they all lie within the right-of-way, each piece of pavement will have its own centerline (except for certain streets with medians; see below). Examples are:

1. Divided highways and other roads with medians. When determining whether or not a road with a median should have separate centerlines for each direction of travel, the rule of thumb I use is that if the median is as wide as that of a typical interstate cross section, I give the street dual centerlines. (I believe this rule is consistent with how roads on USGS topo maps are depicted.) Otherwise, the street is represented with a single centerline. See Figure 7, next page. However, in the situation where the median is relatively wide (but narrower than an interstate), and the pieces of pavement for opposite directions of travel do not parallel each other, the portions that aren't parallel get separate centerlines. These centerlines constitute a one-way pair. See Figure 8, next page.

2. Access roads that branch off from the main traveled roadway. See Figure 9, two pages over.

3. At intersections with large islands, if the island is large enough that the separated pieces of pavement could logically be considered separate traveled ways, each traveled way will have its own centerline. A lot of intersections of interstate ramps with their connecting streets are like this. See Figure 10, two pages over.

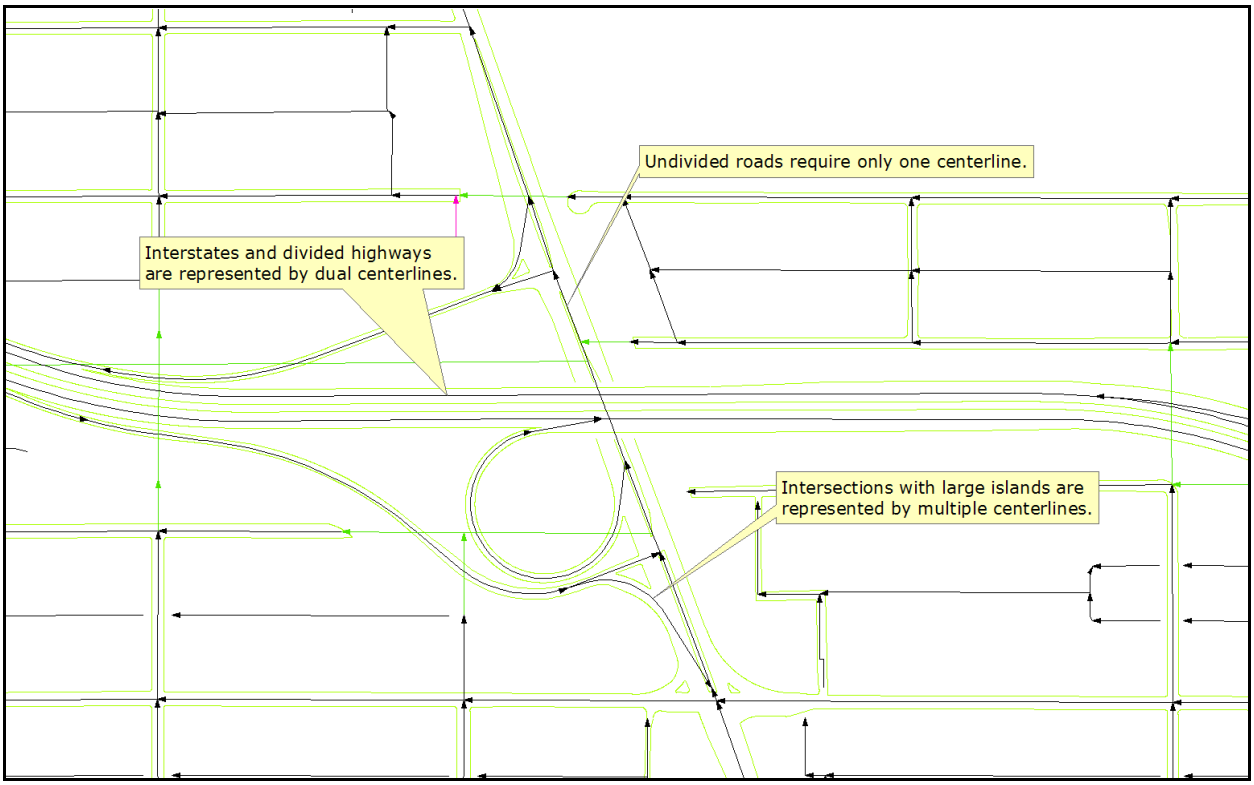


Figure 7 - Example of use of multiple centerlines.

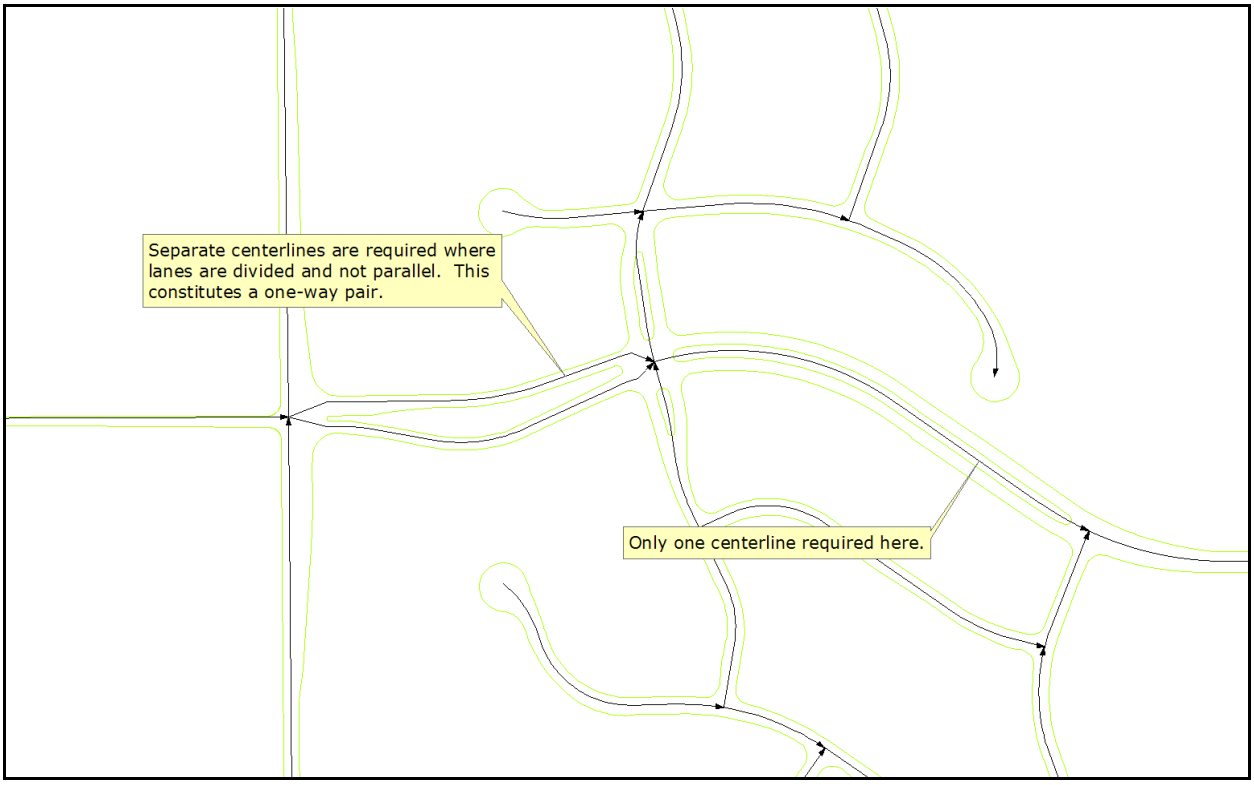


Figure 8 - Another example of dual centerlines.

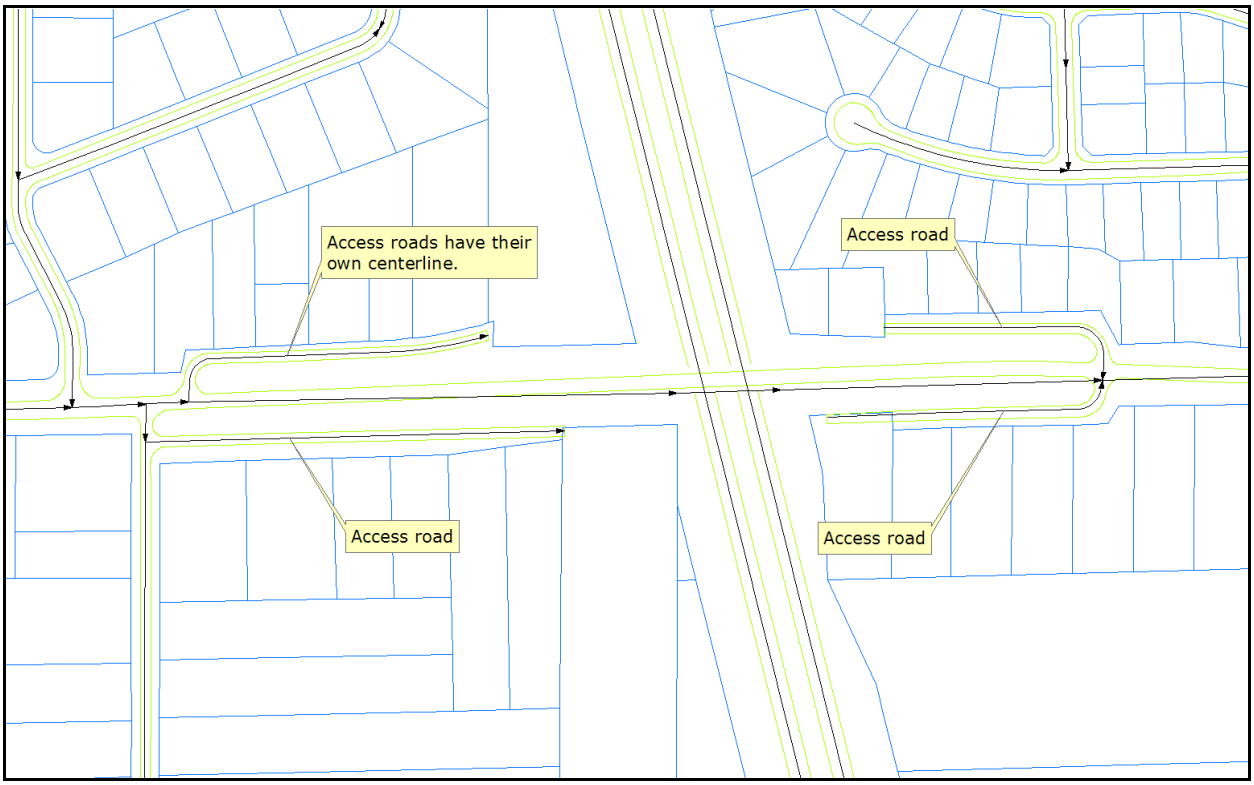


Figure 9 - Parallel access roads are given their own centerlines.

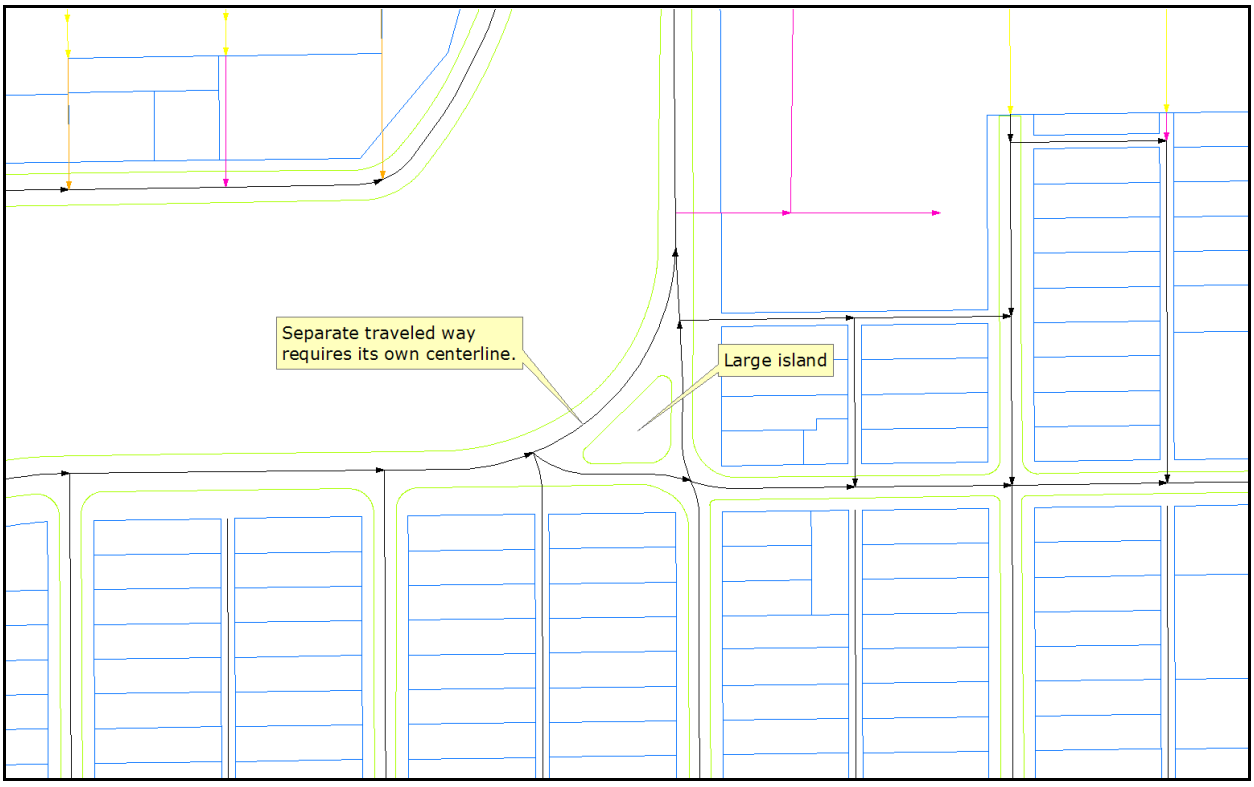


Figure 10 - Another example of the use of multiple centerlines.

4. Another situation where a separate centerline is warranted is where a roadway with a wide median, depicted with dual centerlines, has a turn lane which separates from the main section of pavement. See Figure 11 below.

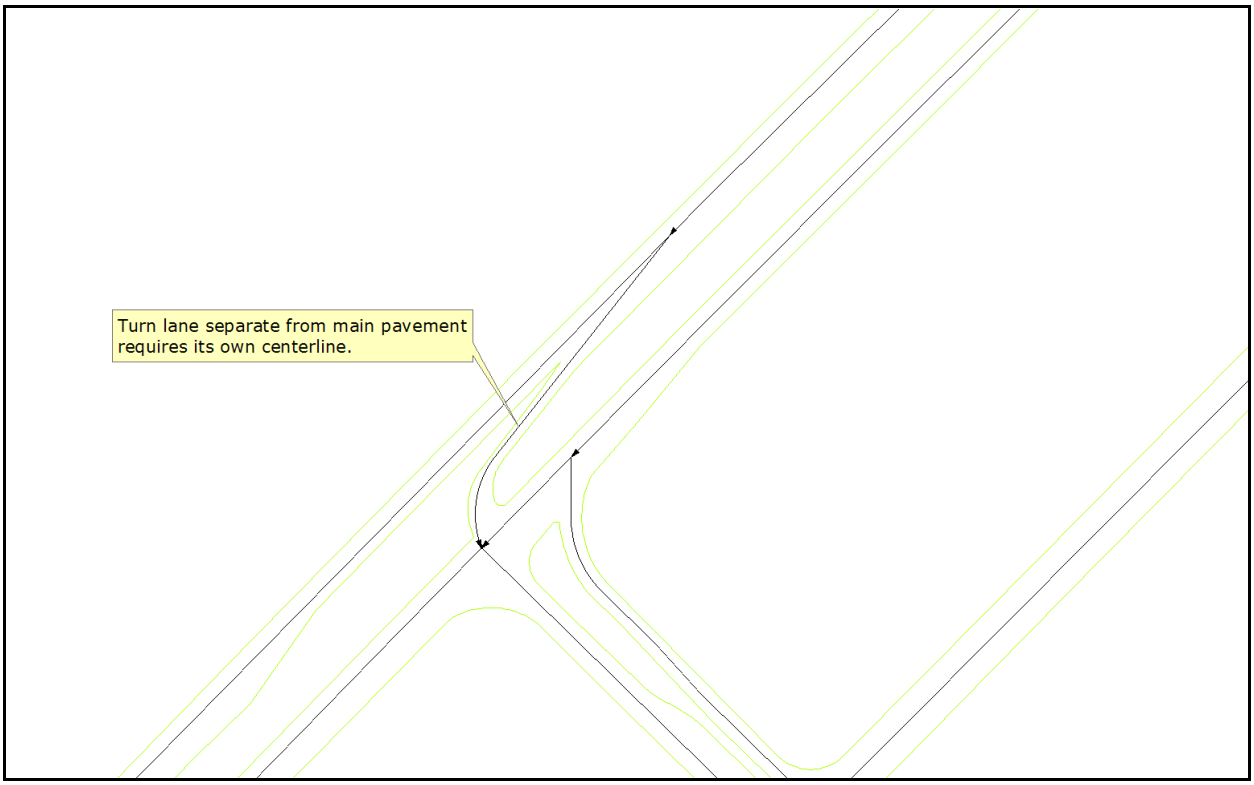


Figure 11 - Example of a turn lane that separates from the main pavement, thus requiring a separate centerline.

In all cases described above where a street has a major piece of pavement and a minor piece, the minor piece will be assigned the same street name as the major piece, unless the minor piece has been explicitly named something different.

Breaking Segments

Segments will only be broken at at-grade intersections of built (improved) streets, or where an attribute changes. They are not broken at overpass intersections, or intersections of non-built streets with other streets. This keeps the number of centerline segments to a minimum, without sacrificing network integrity or quality.

Intersections

A lot of times the centerlines meeting at an intersection don't meet at exactly the same point, assuming the centerlines are all aligned with the center of the right-of-way. Oftentimes, one or more of the streets are offset a certain distance. Thus the concept of a "logical" intersection was created. Logical intersections simplify your street centerline network by allowing you to replace multiple intersections of offset centerlines with one logical intersection. In order to accomplish this, tiny insignificant segments at offset intersections are deleted, and the resulting centerline endpoints

merged together so that they meet at a common point (logical intersection). This simplifies the street network and its maintenance, at the slight cost of compromising the cartographic representation of the centerlines.

A logical intersection will contain only one point of intersection of the centerlines participating in the intersection. A logical intersection can be thought of as all the streets comprising an intersection that are (or would be) controlled by one set of traffic controls (e.g., one traffic light or one set of stop signs), assuming the streets are constructed (built). I also think of a logical intersection as all the streets comprising an intersection from which traffic on any one of the streets could theoretically execute a turn onto any one of the other streets (assuming all the streets allow two-way traffic), without having to yield to traffic at another intersection. This criteria also assumes the streets are built. However, the concept of logical intersections can apply to non-built as well as built streets, because both the criteria are theoretical.

When deciding whether or not to replace multiple offset intersections with a logical intersection, I turn on our aerial photography and try to visualize vehicles negotiating the intersection. I apply the criteria of traffic controls and vehicular movement noted above to make a determination. Admittedly, sometimes the choice is somewhat arbitrary. For the purposes of logical intersections, alleys are treated just like streets.

If it is determined that an offset intersection will be treated as one logical intersection, one or more of the centerlines participating in the intersection will be bent, so that all the centerlines meet at a single point. For each logical intersection, I try to pick a point of intersection so as to keep the number of bent centerlines to a minimum. This is because, even though our centerlines are not survey accurate, I like to keep the amount of distortion from the center of right-of-way to a minimum. This is so someone can take measurements from the centerline to an offset point, and trust that identical measurements on the map and in the field will be roughly equal. Typically, the extent of the centerline distortion on the map is limited to the right-of-way limits within the intersection (or edge of pavement, if no right-of-way).

When bending centerlines at an intersection, I usually prefer to use circular arcs for the bent portion. This is purely for aesthetic reasons. Figure 12 on the next page is an example of an offset intersection converted to a logical intersection.

If the opposing legs of an offset intersection are sufficiently far enough apart (using the above criteria), the centerlines will not be altered, and the intersection will be treated as two separate logical intersections (Figure 13, next page).

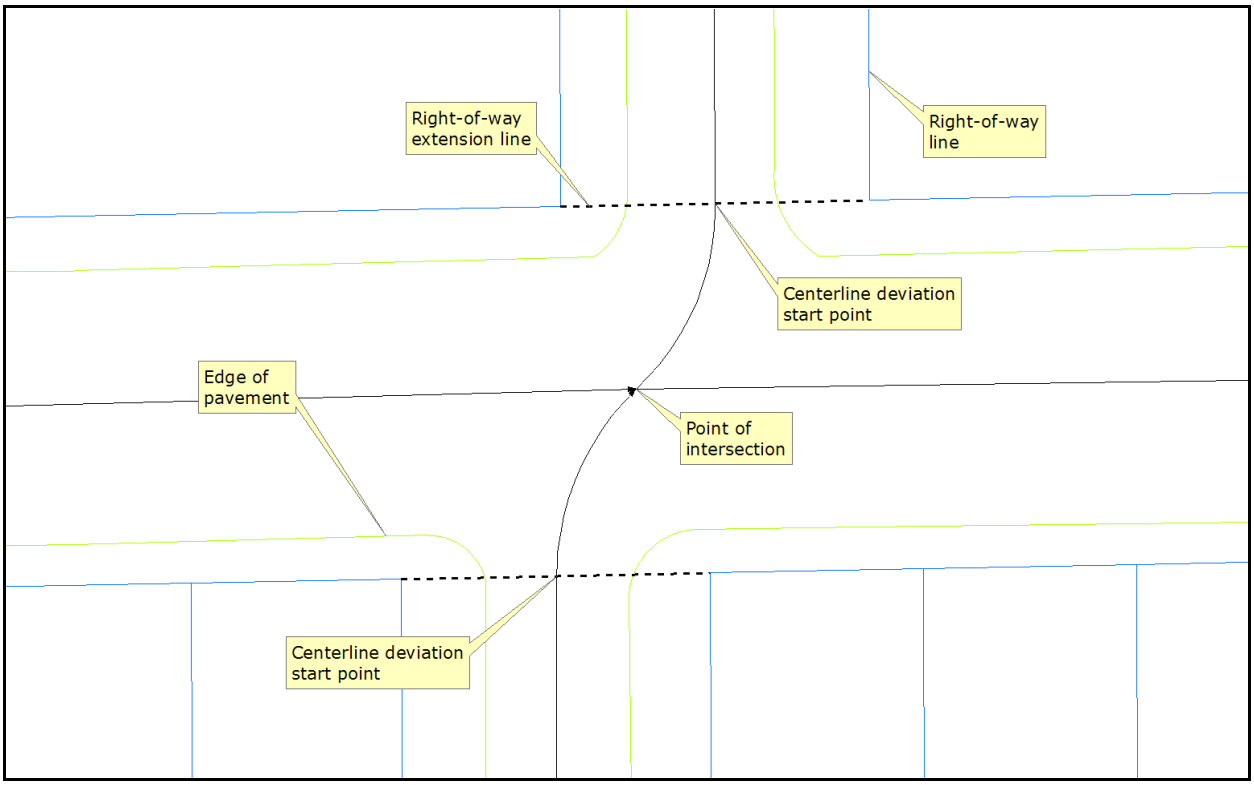


Figure 12 - A logical intersection. A driver traveling north could conceivably negotiate the intersection in one movement, without having to stop (following the alignment of the curved centerlines).

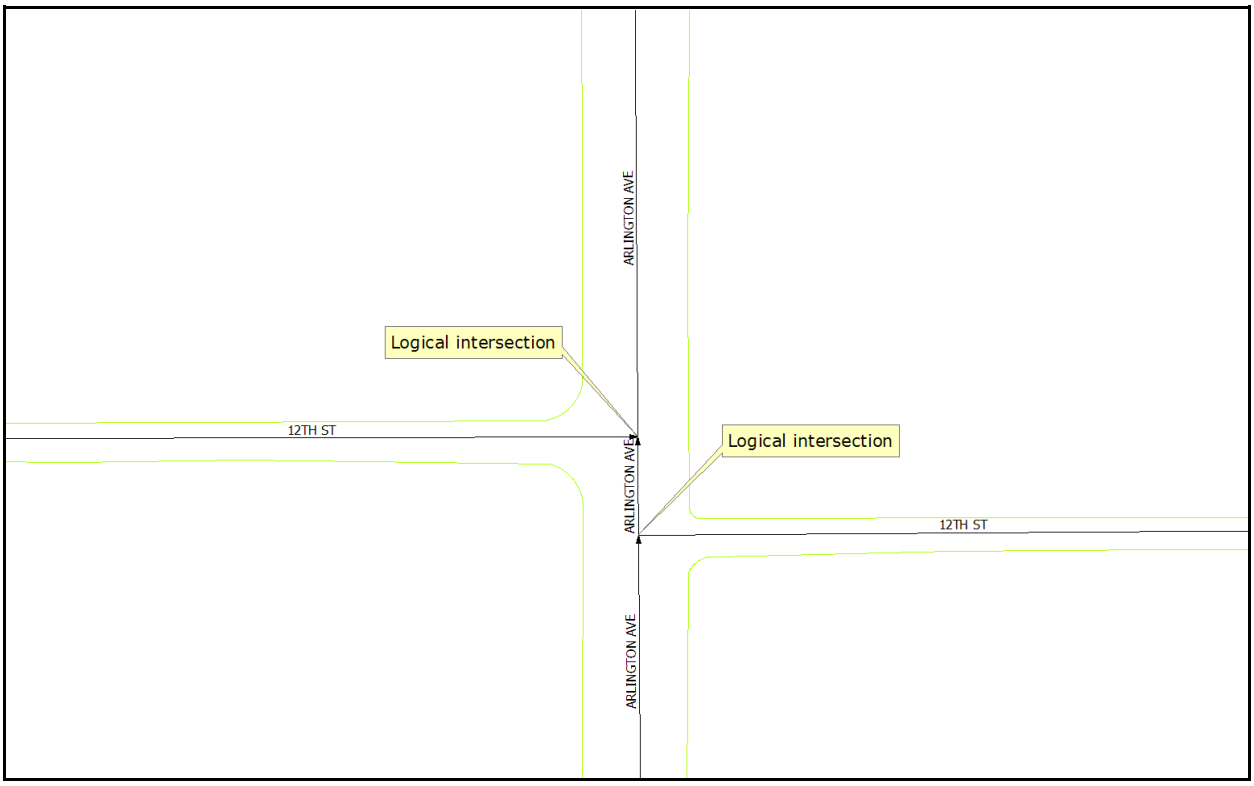


Figure 13 - A driver traveling west on 12th St. could turn right onto Arlington Ave., and then conceivably have to wait on oncoming traffic, before being allowed to turn west again onto 12th Street.

If more than one of the streets entering an intersection are offset, then the choice of a suitable point of intersection for the logical intersection is a bit more complicated. In this case I choose a point such that the bending occurs within a selected radius of that point, and opposing segments are bent symmetrically, if possible. Usually it is possible to locate the preferred point geometrically, using some temporary sketch lines. Figure 14 is an example of this type of intersection.

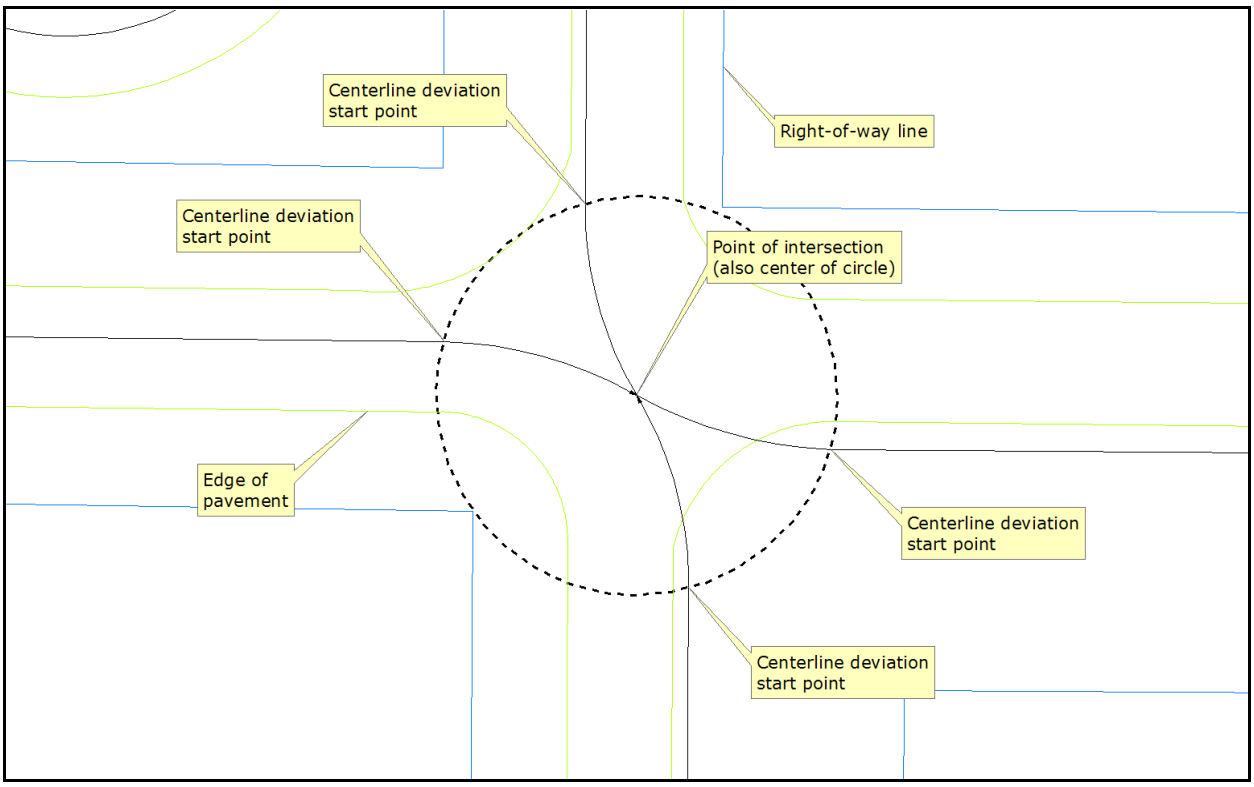


Figure 14 - A double-offset intersection. Note that the centerlines are centered on the right-of-way, not the edge of pavement.

If an intersection is a three-way intersection where the centerlines theoretically do not meet, and two of the legs belong to the same street, then those legs will not be altered, and the other street will be warped to intersect the intersection of the first two.

Sometimes the configuration of an intersection is such that it is not readily apparent how the centerlines should be bent. In these cases I try to give preference to the street with the higher traffic volume, or to the most common traffic movement through the intersection (meaning the streets with less volume or less common movements would be bent). Sometimes the decision of where to place the point of intersection boils down to whichever arrangement produces the most aesthetically pleasing alignment. See Figures 15 and 16 next page.

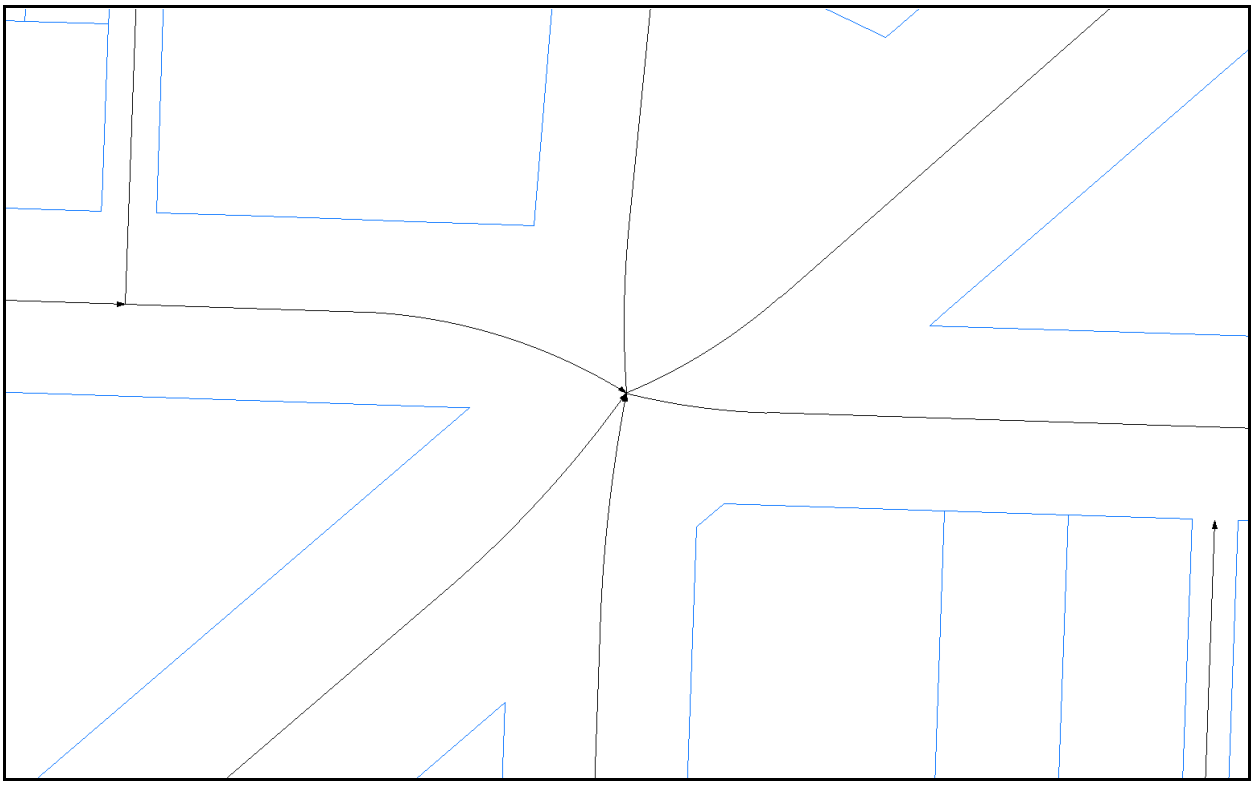


Figure 15 - Logical intersection with all segments bent to intersect at common node.

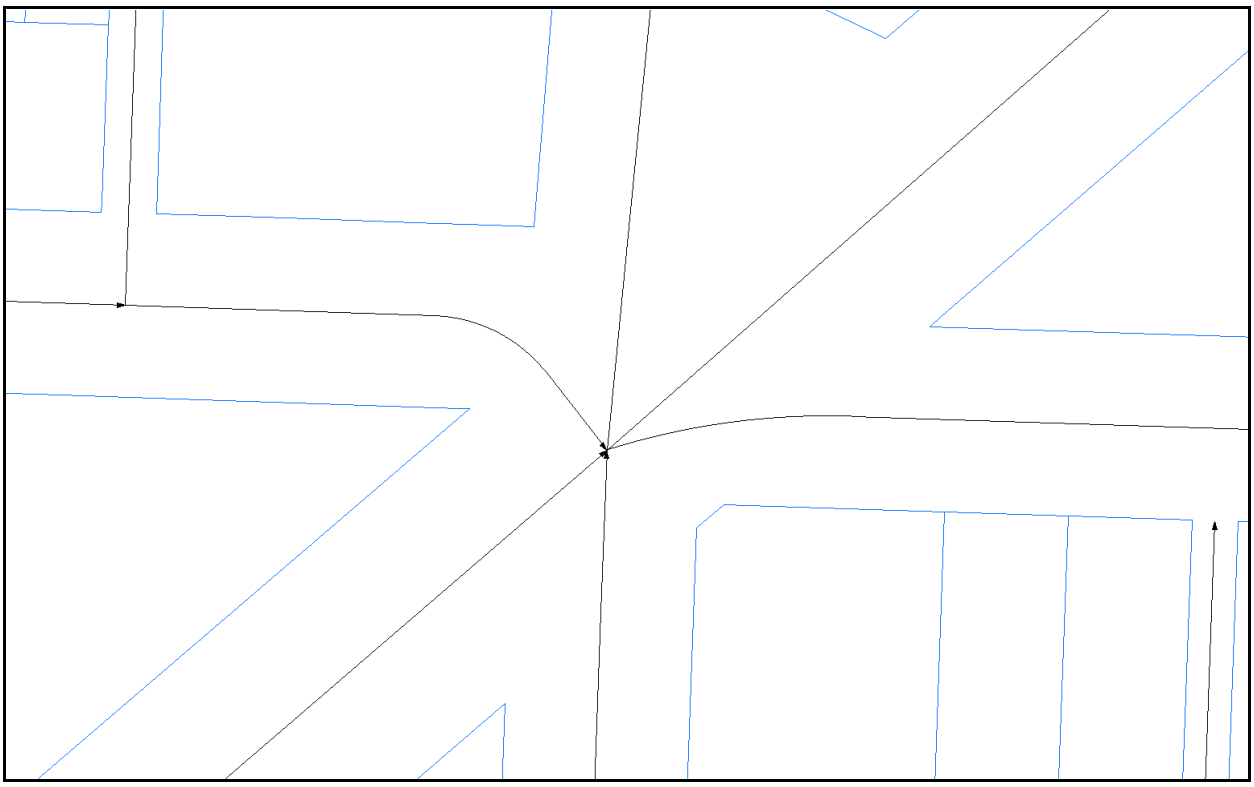


Figure 16 - Logical intersection with only two segments bent — better arrangement, in my opinion.

There are cases where, due to the particular configuration of an intersection, the centerline bending is not confined to the right-of-way limits inside the intersection. Sometimes this is done to avoid warping the centerlines so much that the result appears awkward. In these cases, the centerline may be drawn so that it follows the center of pavement through the intersection. See Figure 17.

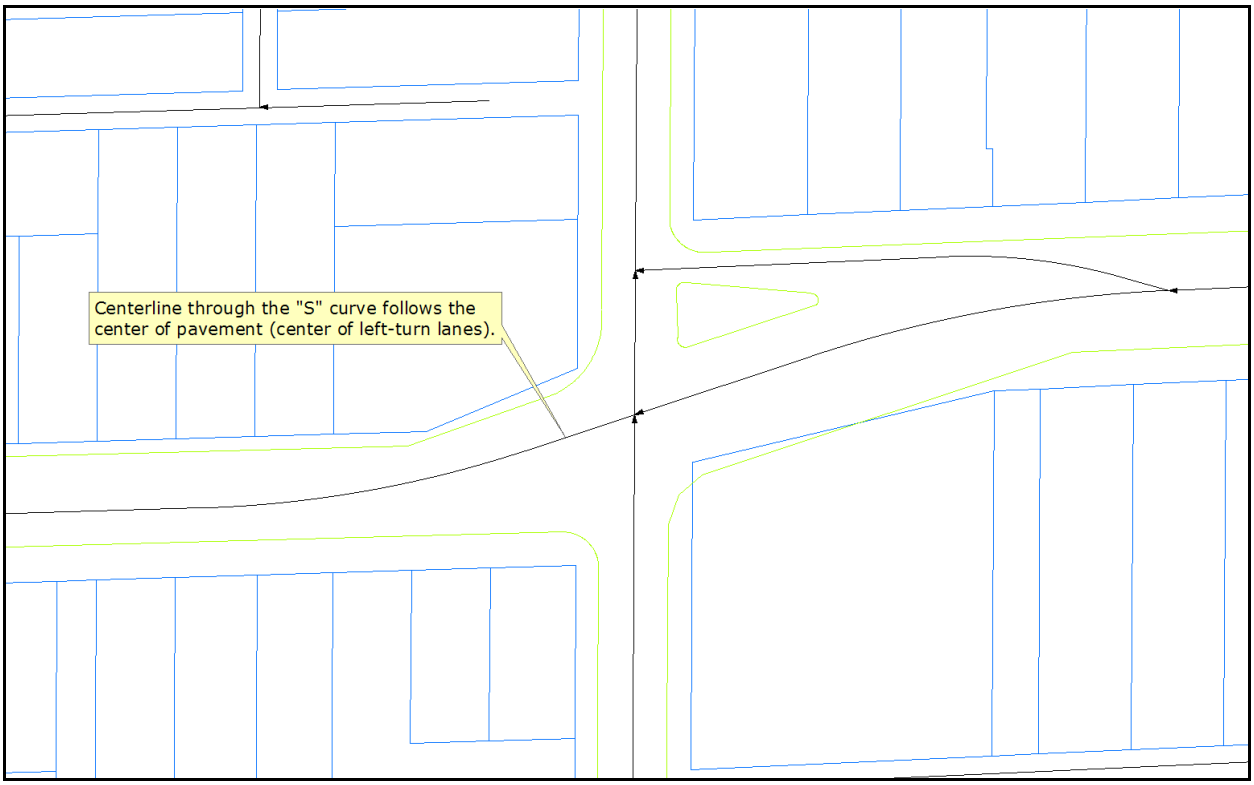


Figure 17 - Example of offset intersection where centerline warping is not confined to the right-of-way limits of the intersection.

Intersections of streets with divided highways are a special case. Because divided highways have a separate centerline for each set of lanes, the configuration could be considered two separate intersections. However, I consider this one logical intersection, so I came up with a special arrangement of centerlines for these intersections so that they all meet at the same point. These types of intersections constitute one of the few exceptions to the rule against overlapping centerlines. Consequently the areas of centerline overlap in these intersections are marked as exceptions in the topology layer. See Figures 18 and 19, next page, to see what these intersections look like.

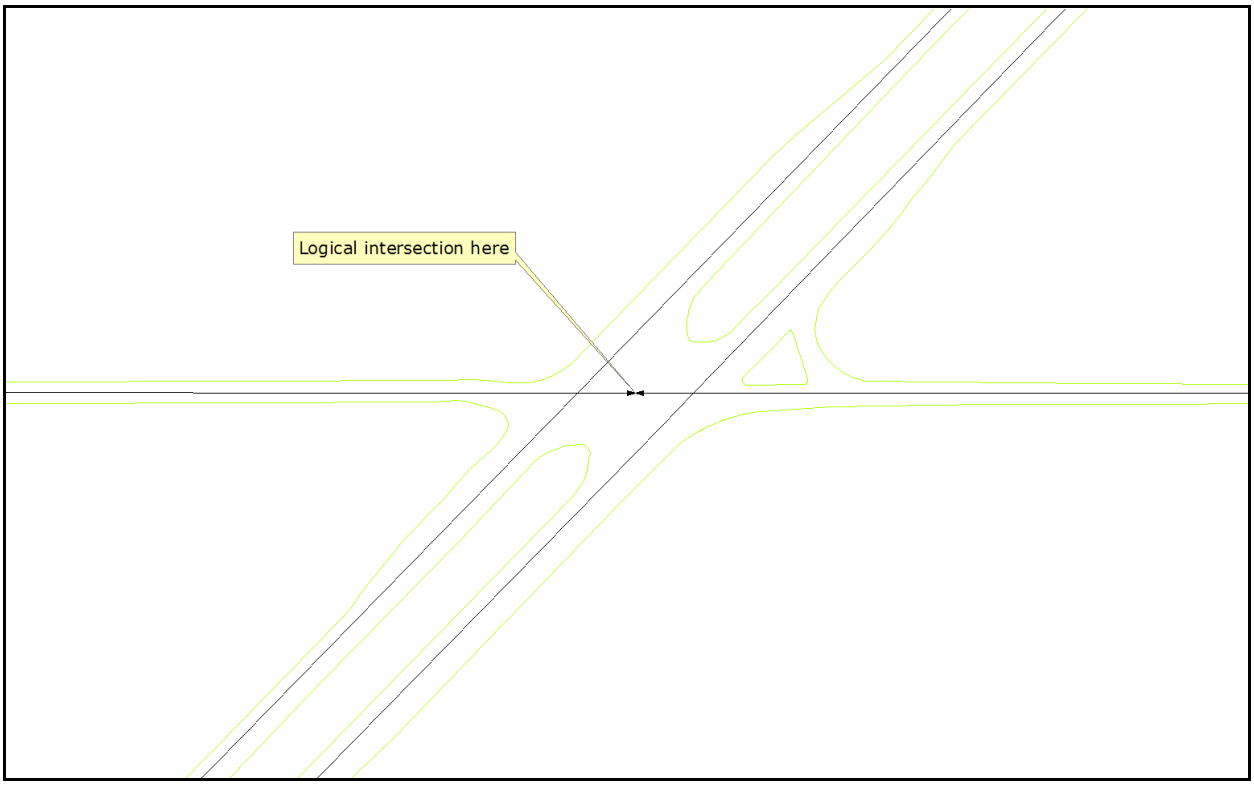


Figure 18 - An intersection of a street with a divided highway. This is considered one logical intersection, where all the centerlines entering the intersection meet.

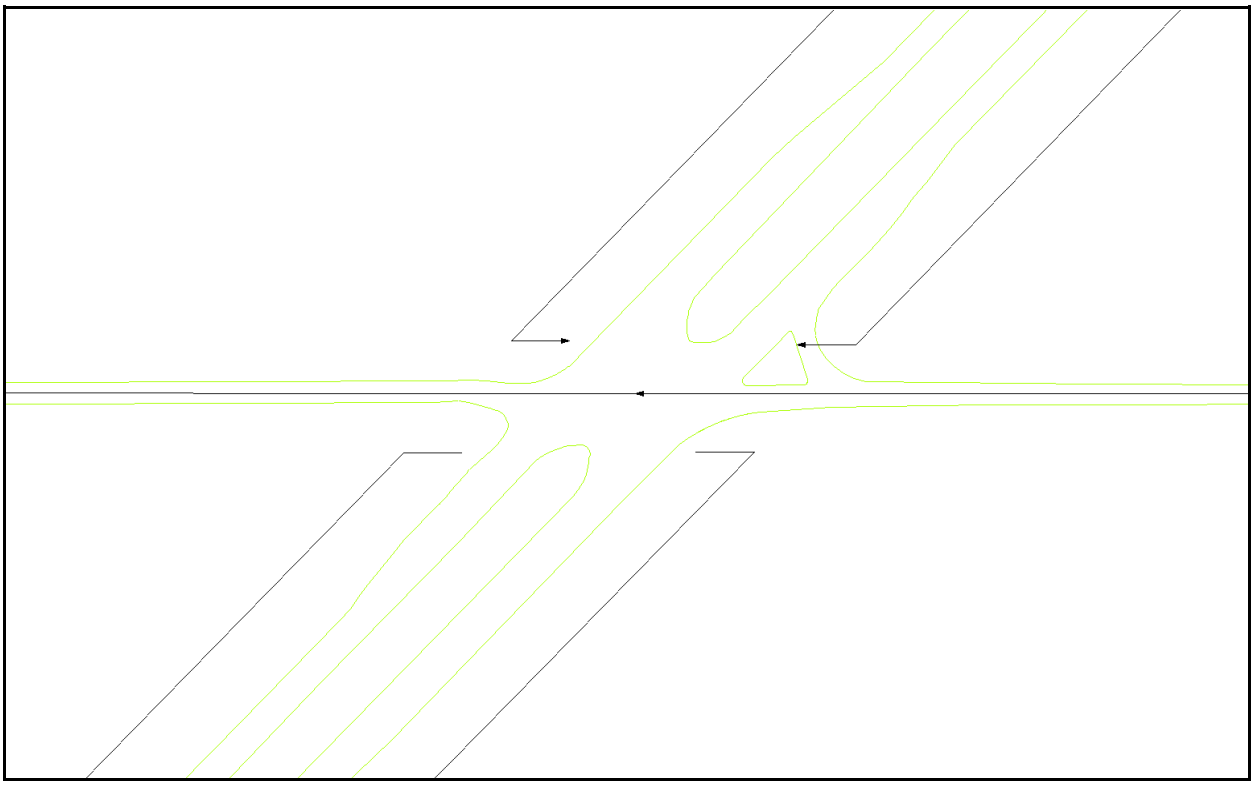


Figure 19 - This is what the individual centerlines look like. The centerlines for the divided highway have been moved away from the intersection in this illustration, in order to show their geometry.

If any of the segments of a logical intersection are not traversable (non-built), those segments will not be bent to meet at the common point of intersection in the logical intersection. Instead, those segments will retain their original platted alignment. (Since non-traversable segments don't participate in routing applications, there is no need for physical continuity between them and the rest of the network, and thus no need to alter them.) See Figure 20.

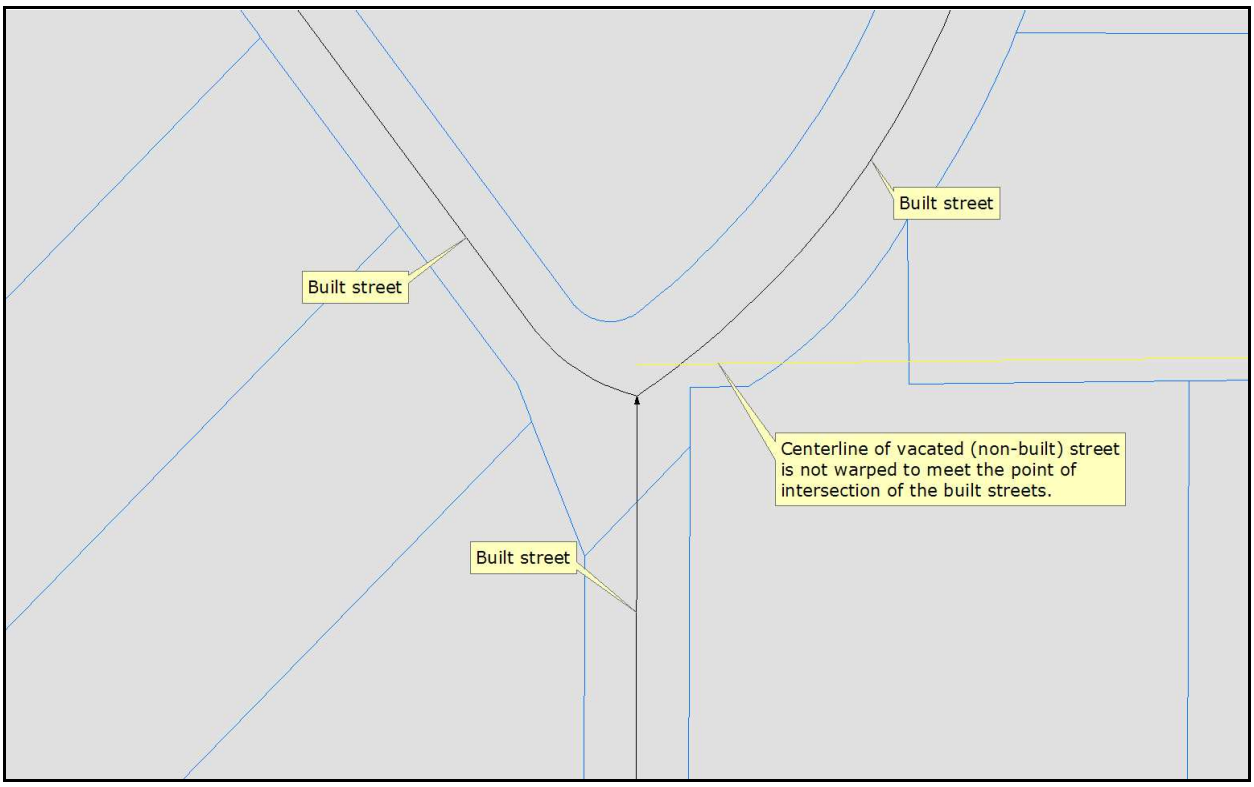


Figure 20 - Example of logical intersection involving built and non-built streets.

Direction

Except for interstates and ramps, all centerlines within Marion County will be digitized in the direction of increasing addresses, regardless of whether or not the addresses increase in the proper direction (according to address grid). Interstate and ramp centerlines will be digitized in the direction of traffic flow.

Loops

A cul-de-sac shall terminate in a loop if it contains a non-traversable island in the center, otherwise it shall consist solely of a dangling centerline. If it ends in a loop, the loop shall be split in the back, to accommodate routing algorithms that can't handle a closed loop. If the cul-de-sac contains even addresses on one side of the street and odd addresses on the other, the split will be placed even with the junction of the even and odd ranges; otherwise, it will be split approximately at its midpoint. See Figure 21 next page.

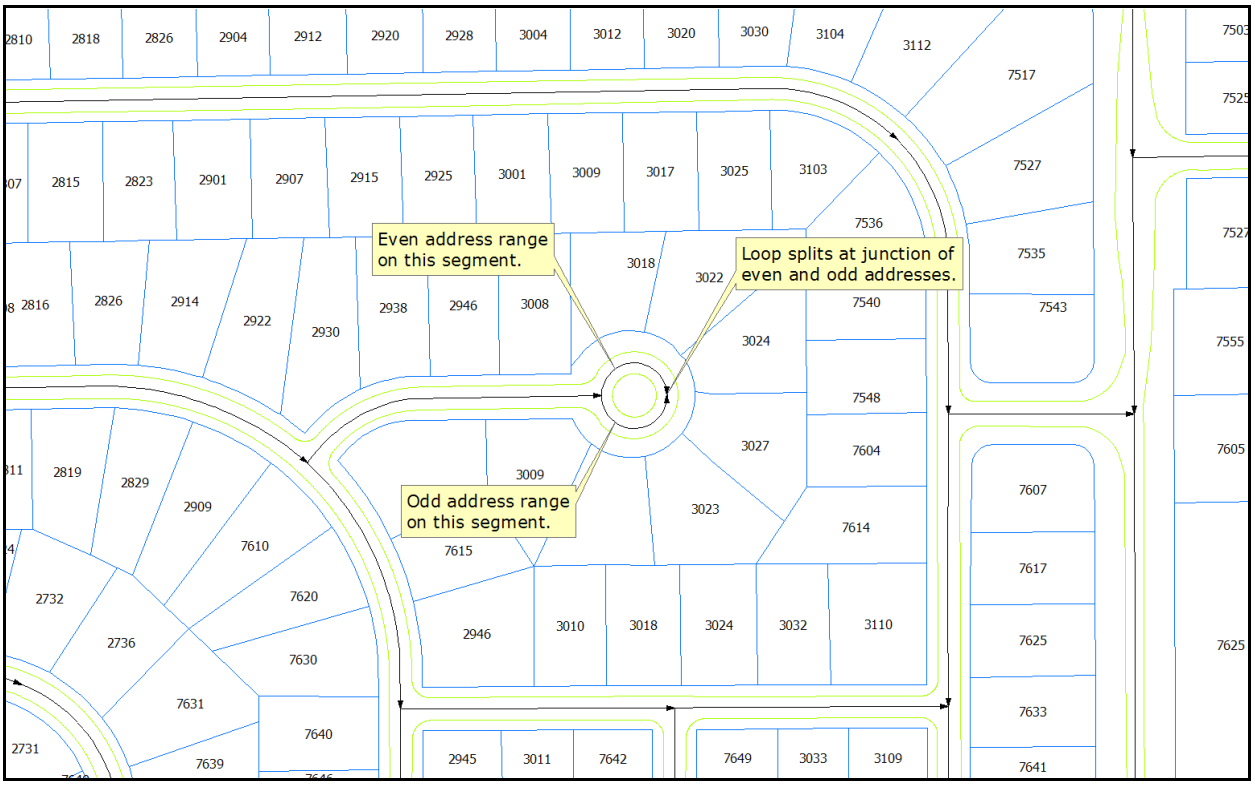


Figure 21 - Typical cul-de-sac with island.

Overlapping centerlines

Centerlines don't overlap, with four exceptions. The first exception is for streets with multiple address ranges in the same block. In these situations, I use identical, overlapping centerlines, one for each set of address ranges. This allows both sets of ranges to be geocoded. An example of this is 25th Ave./Perkins Ave. in Beech Grove, where parcels on one side of the street are addressed off the Indianapolis address grid, and the parcels on the other side of the street are addressed off the Beech Grove address grid. See Figure 22, next page.

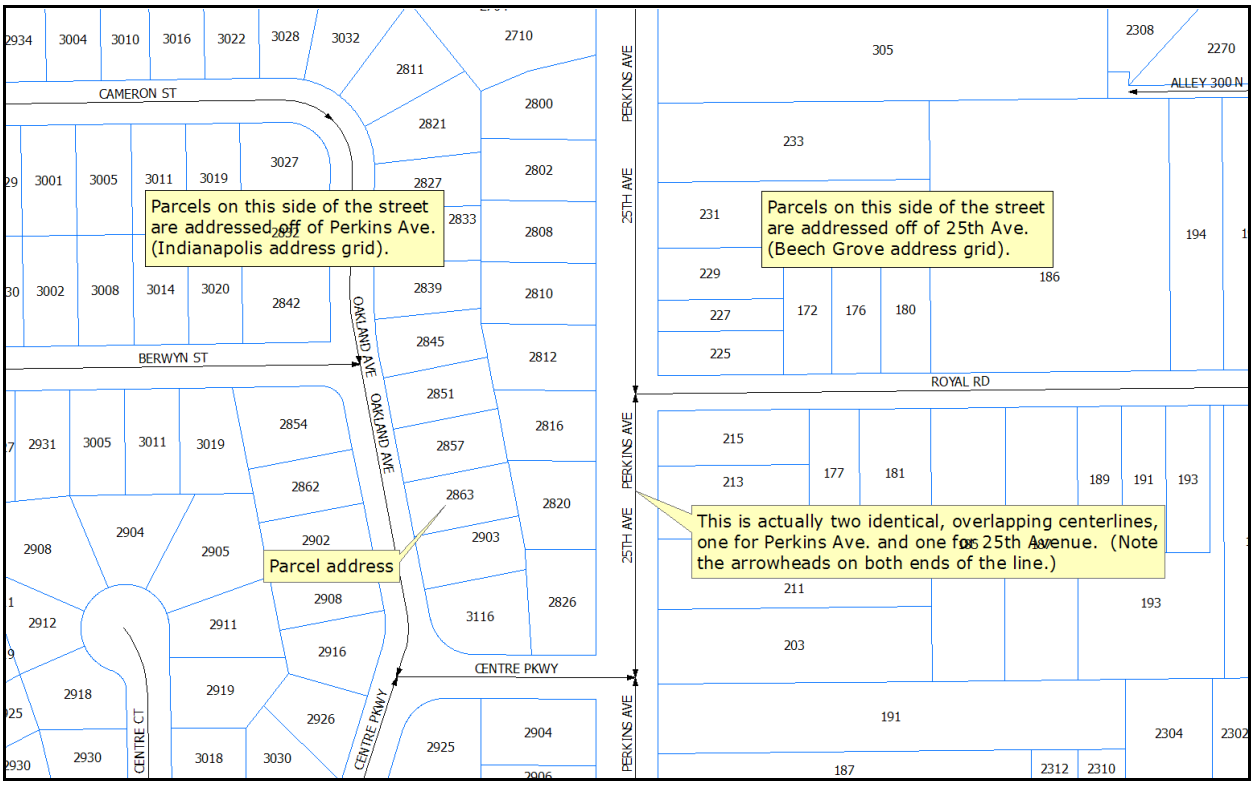


Figure 22 - Example of a street having coincident centerlines, because it is addressed with multiple address ranges.

The second exception for overlapping centerlines is for dead-end streets where the addresses increase on one side of the street, and decrease on the other side of the street (going the same direction). Again, here there will be identical, overlapping centerlines, digitized in opposite directions, so that the addresses can be geocoded properly. See Figure 23, next page.

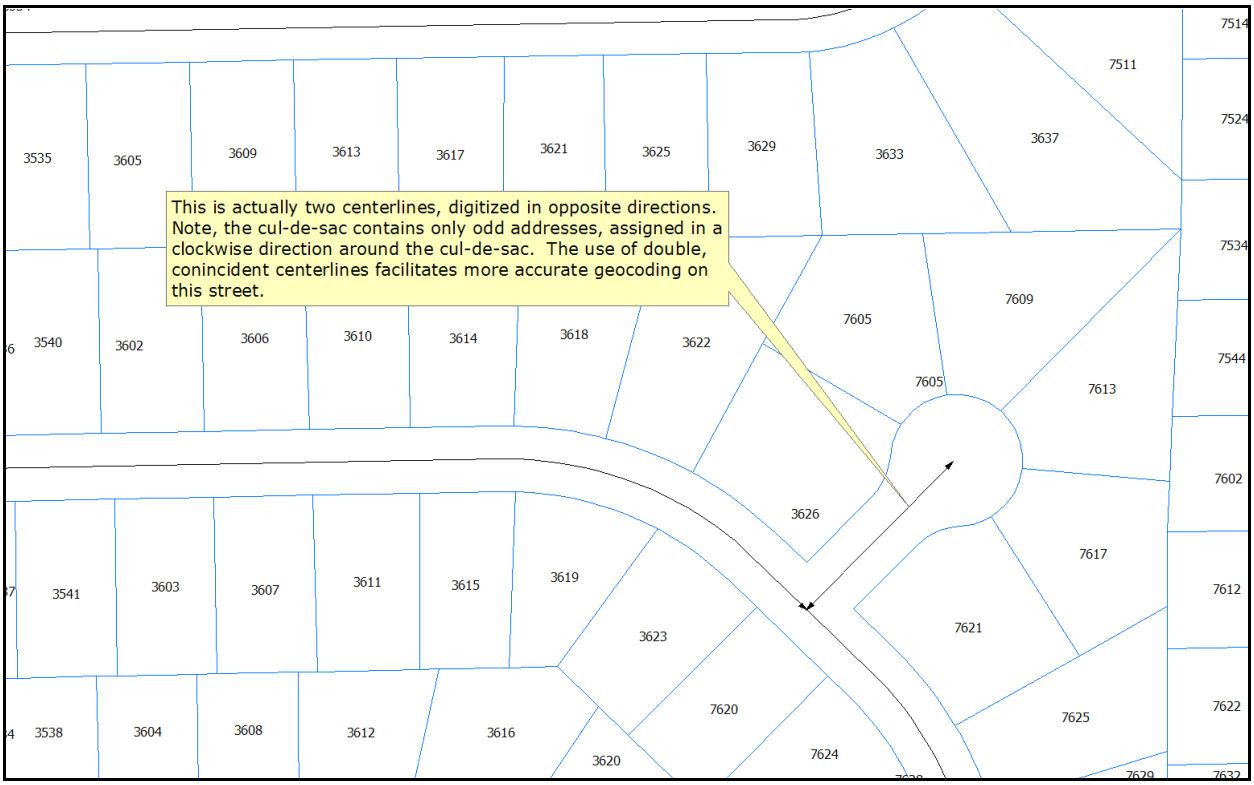


Figure 23 - A cul-de-sac modeled with coincident centerlines. Note the double arrowheads.

The third exception is for intersections of streets with divided highways, as discussed earlier and shown in Figures 18 and 19.

The last exception pertains to certain intersections where two streets follow the same alignment for a very short distance. This is a relatively uncommon occurrence, but it does happen. See Figures 24 and 25 on the next page for an example.

In all cases of overlapping centerlines, the centerlines are identified with the words DOUBLE CENTERLINE in their Remarks attribute (see page 51). They are also shown as exceptions in the topology layer (see page 96). These two methods allow overlapping centerlines to be easily identified, since they are not apparent just by looking at them.

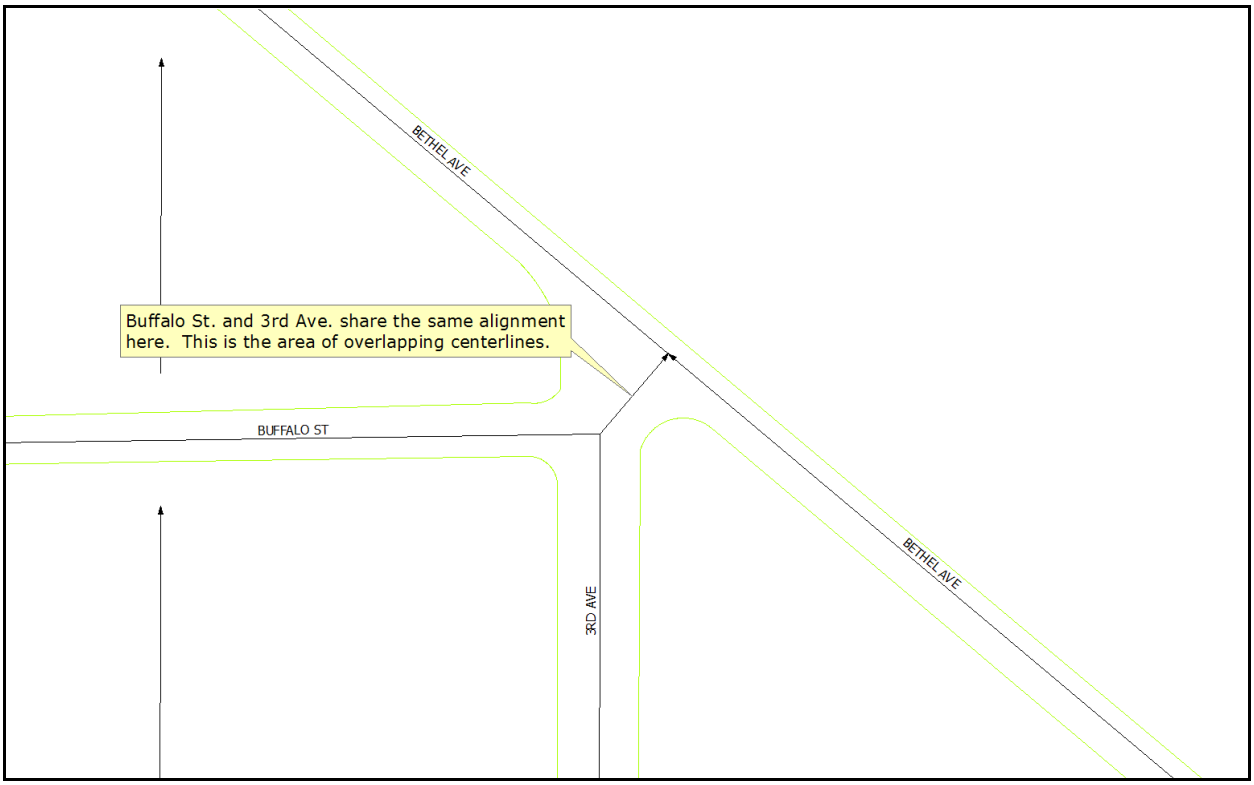


Figure 24 - A typical intersection where two streets share the same alignment for a very short distance.

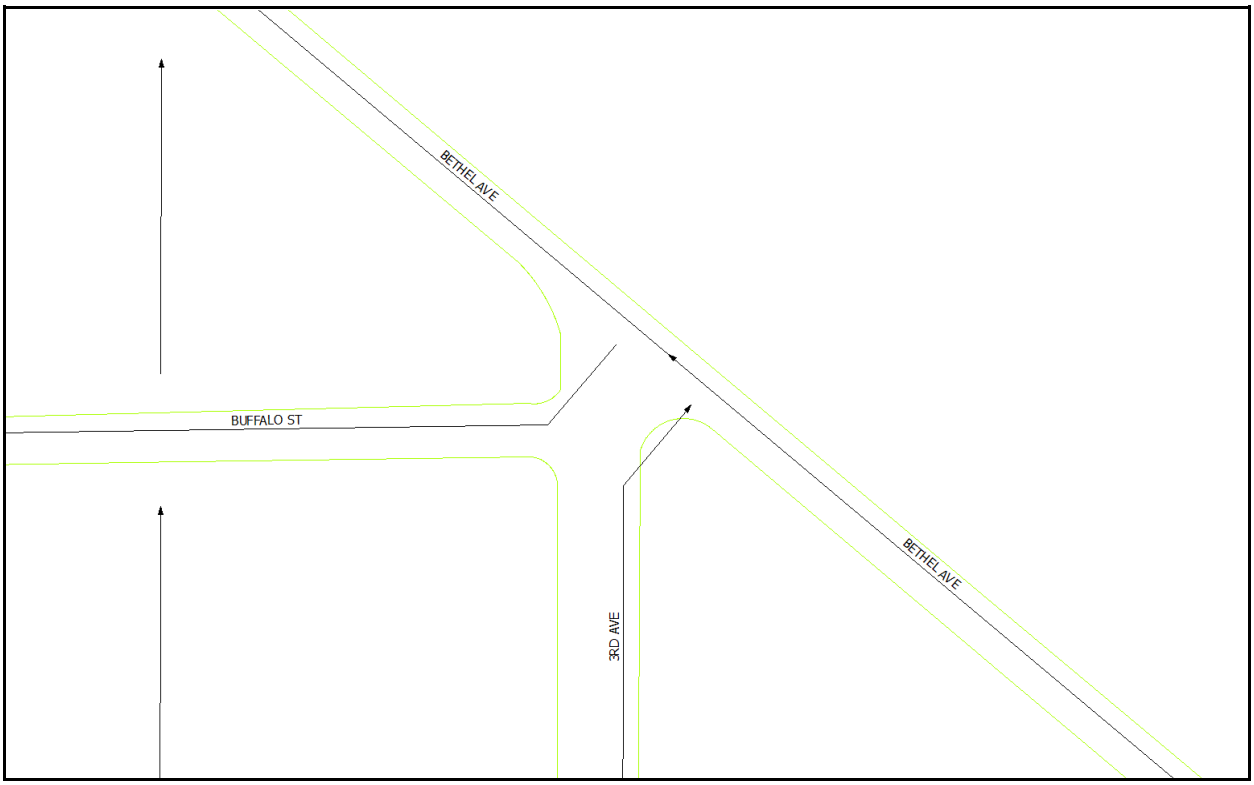


Figure 25 - Here the centerlines for Buffalo St. and 3rd St. have been moved apart, in order to show their respective alignments.

Attributes

All alphabetical values will be entered in all upper case. Street names and portions thereof will follow the *Address Guidelines and Standards* published by DMD, together with "unofficial" amendments and policies developed by the Master Address Database Committee of the ISA GIS staff. Note that no attributes pertaining to pavement characteristics are maintained on the centerlines; these are maintained in a separate *Pavement Management System* database, with the goal that this system will eventually be linked to the centerlines.

The City of Indianapolis created an object model to define the relationships between the street centerlines and associated layers. Since we decided we were going to create a trail centerline layer in addition to the street centerline layer, we looked at the attributes that we thought they should both have in common. Based on this observation, an abstract feature class was created to contain all the attributes common to both layers. From this abstract class, feature classes for both the street and trail centerlines were created, which inherit the common attributes. In addition, each feature class also contains attributes unique to its own class.

(Note: As of this writing [October 2007], the trails layer has not been implemented.)

Centerlines in the *Streets* layer are required to have all attributes populated (even if some of them are null). The only attributes required to be populated for centerlines in the *StreetsOOC* layer are COUNTY_LEFT and COUNTY_RIGHT, OPER_STATUS, MAINT_JURIS, DATE_CREATED, STREET_NAME, and FULL_STNAME.

Some of the attributes have domains defined for them. For all the domains, the Split Policy is set to "Duplicate," and the Merge Policy is set to "Default Value."

On the following two pages is a table listing all the attributes of the street centerlines, in the order they appear in the attribute-editing window, and their definitions. All the attributes from the beginning through COORDDIR are inherited from the abstract centerlines class; the ones after that are unique to the street centerlines. This explains why some of the attributes don't seem to be grouped logically.

Following the table is a detailed description of each attribute and its domain, if it has one.

Centerline Attributes

Field Name	ORACLE Type	Length	Allow Null Values	Default Value	Domain Name	Precision	Scale
CENTERLINE_TAG	Long Integer	-	No	0	-	9	-
COMPKEY	Long Integer	-	Yes	-	-	9	-
COMPTYPE	Long Integer	-	Yes	-	-	5	-
L_ADD_FROM	Long Integer	-	Yes	0	domAddrRange	5	-
L_ADD_TO	Long Integer	-	Yes	0	domAddrRange	5	-
R_ADD_FROM	Long Integer	-	Yes	0	domAddrRange	5	-
R_ADD_TO	Long Integer	-	Yes	0	domAddrRange	5	-
MILE_FROM	Float	-	Yes	-	-	6	2
MILE_TO	Float	-	Yes	-	-	6	2
L_ZIP	Text	5	Yes	-	domZipCode	-	-
R_ZIP	Text	5	Yes	-	domZipCode	-	-
L_TRACT	Long Integer	-	Yes	-	-	6	-
R_TRACT	Long Integer	-	Yes	-	-	6	-
COUNTY_LEFT	Text	20	No	MARION	domCounty	-	-
COUNTY_RIGHT	Text	20	No	MARION	domCounty	-	-
CITY_LEFT	Text	20	Yes	INDIANAPOLIS	domCity	-	-
CITY_RIGHT	Text	20	Yes	-	domCity	-	-
TWP_LEFT	Text	20	Yes	-	domTownship	-	-
TWP_RIGHT	Text	20	Yes	-	domTownship	-	-
OPER_STATUS	Text	15	No	-	domOperStatus	-	-
MAINT_JURIS	Text	16	No	DEVELOPER	domMaintJuris	-	-
OLD_NAME	Text	50	Yes	-	-	-	-
NO_ADDR	Text	5	Yes	-	domNoAddr	-	-
ADDRPROB	Text	5	No	91N	domProb	-	-
REMARKS	Text	70	Yes	-	-	-	-
DATE_CREATED	Date	-	Yes	-	-	-	-
DATE_MOVED	Date	-	Yes	-	-	-	-
DATE_CHANGED	Date	-	Yes	-	-	-	-
COORDINATE	Double	-	Yes	-	-	16	0
COORDDIR	Text	70	Yes	-	domDir	-	-
TFARE	Text	27	Yes	COLLECTOR/LOCAL STREET	domTFare	-	-
STRCLASS	Text	1	Yes	D	domStreetClass	-	-
STRLEVEL	Short Integer	-	Yes	0	domLevel	2	-
STRSUBTYPE	Long Integer	-	No	7	-	9	-
SPD_LIM	Short Integer	-	Yes	-	domSpdLim	2	-
SECONDS	Short Integer	-	Yes	-	-	3	-
SPD_LIM_ORD	Text	25	Yes	-	-	-	-
ONE_WAY	Text	2	Yes	-	domOneWay	-	-
ONE_WAY_DIR	Text	5	Yes	-	domOneWayDir	-	-
ONE_WAY_ORD	Text	25	Yes	-	-	-	-
NAME_CHANGE	Text	50	Yes	-	-	-	-
VACATED	Text	50	Yes	-	-	-	-
PRE_DIR	Text	1	Yes	-	domDir	-	-
STREET_NAME	Text	20	Yes	-	-	-	-
STREET_TYPE	Text	4	Yes	-	domStreetType	-	-
SUF_DIR	Text	1	Yes	-	domDir	-	-

Centerline Attributes

FULL_STNAME	Text	30	Yes	-	-	-	-
STR_LABEL	Text	30	Yes	-	-	-	-
ALIAS_FULL_STNAME	Text	50	Yes	-	-	-	-
ADDRESSING_GRID	Text	15	Yes	-	-	-	-
WEIGHT_LIM	Long Integer	9	Yes	-	-	-	-
WEIGHT_LIM_ORD	Text	25	Yes	-	-	-	-
BLOCK_ID	Long Integer	10	Yes	-	-	10	-
DATE_ACCEPTED	Date	-	Yes	-	-	-	-

CENTERLINE_TAG - The Tag number shall be a unique number used to identify individual centerline segments. Tag numbers are assigned sequentially. In the beginning, when Tag numbers were first assigned, we started with the number 100000, and the numbers increased from there. However, for a short while several years ago, we had an intern who did a little of the maintenance on the centerlines, and he was instructed to number all new centerlines he created starting from 90000. For this reason, we now have a few centerlines numbered in this range.

Initially, Tag numbers were assigned manually. Then a macro was developed to assign new numbers automatically, when the centerlines were still maintained as a coverage. Now new numbers are assigned automatically by a custom centerline attribute editor tool we had developed by a consultant. (The official name of the tool is *CMFI*, or "Custom MAD Feature Inspector." See page 133 for more on this tool.) Numbers have currently surpassed the 180000 figure.

When a centerline is split in two, one of the resulting centerlines retains the original tag number, and the other is assigned a brand new number. (See DATE_CREATED attribute, page 51. See also Figures 26 and 27, next page.) When a tag number is retired, it is never reused. An example of this is when two centerlines are merged. One of the original centerline tags is retained, and one is retired. See Figures 28 and 29, two pages over.

COMPKEY - This field was created to store a value that Hansen's Infrastructure Management System (IMS) program uses to identify street segments. Our Department of Public Works uses IMS for its facilities' management. We had planned to link the centerlines to this database, using this field, in combination with the *COMPTYPE* field. So far this field has yet to be used.

COMPTYPE - This field was created to store a value that Hansen's IMS program uses to identify street segments. So far this has yet to be used.

L_ADD_FROM, L_ADD_TO - The left "from" and "to" address range numbers. Values can be the same value for very short segments. The domain constrains the values to be between 0 and 15000, inclusive.

R_ADD_FROM, R_ADD_TO - The right "from" and "to" address range numbers. Values can be the same value for very short segments. The domain constrains the values to be between 0 and 15000, inclusive.

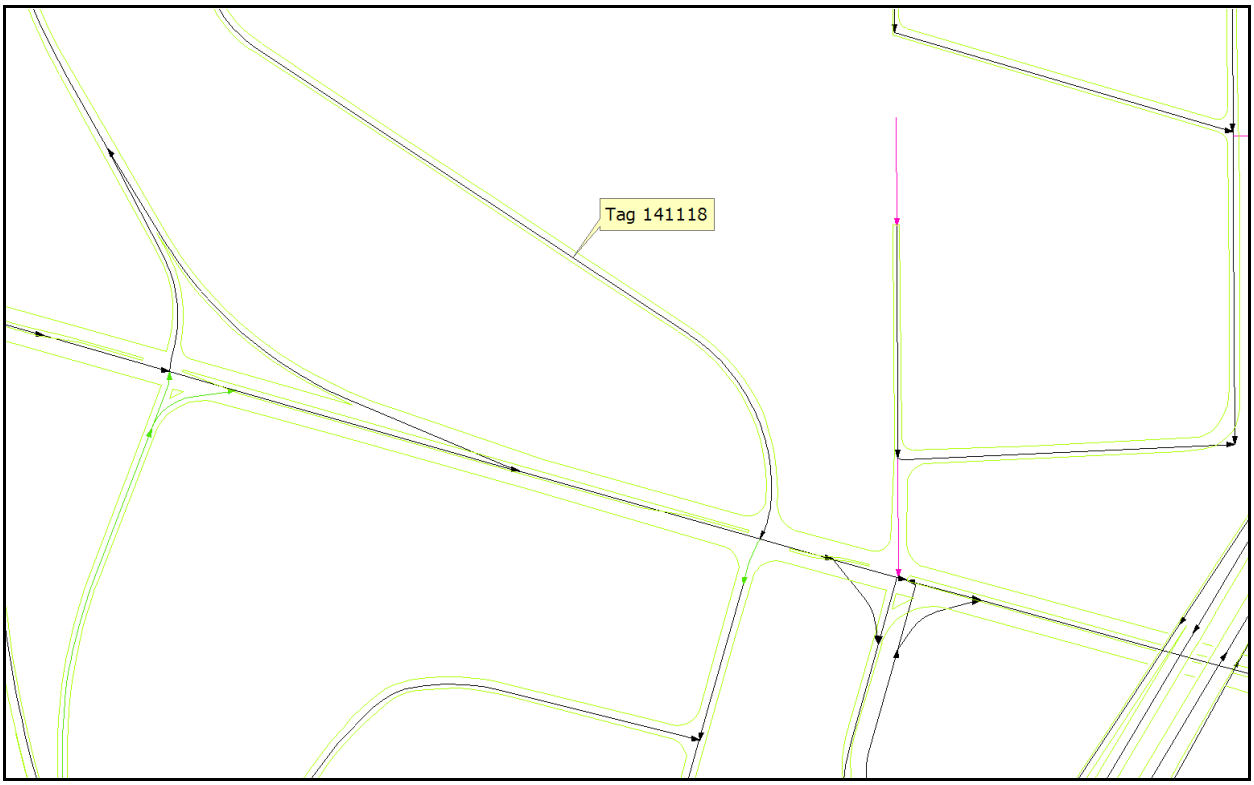


Figure 26 - Shortridge Rd. before realignment.

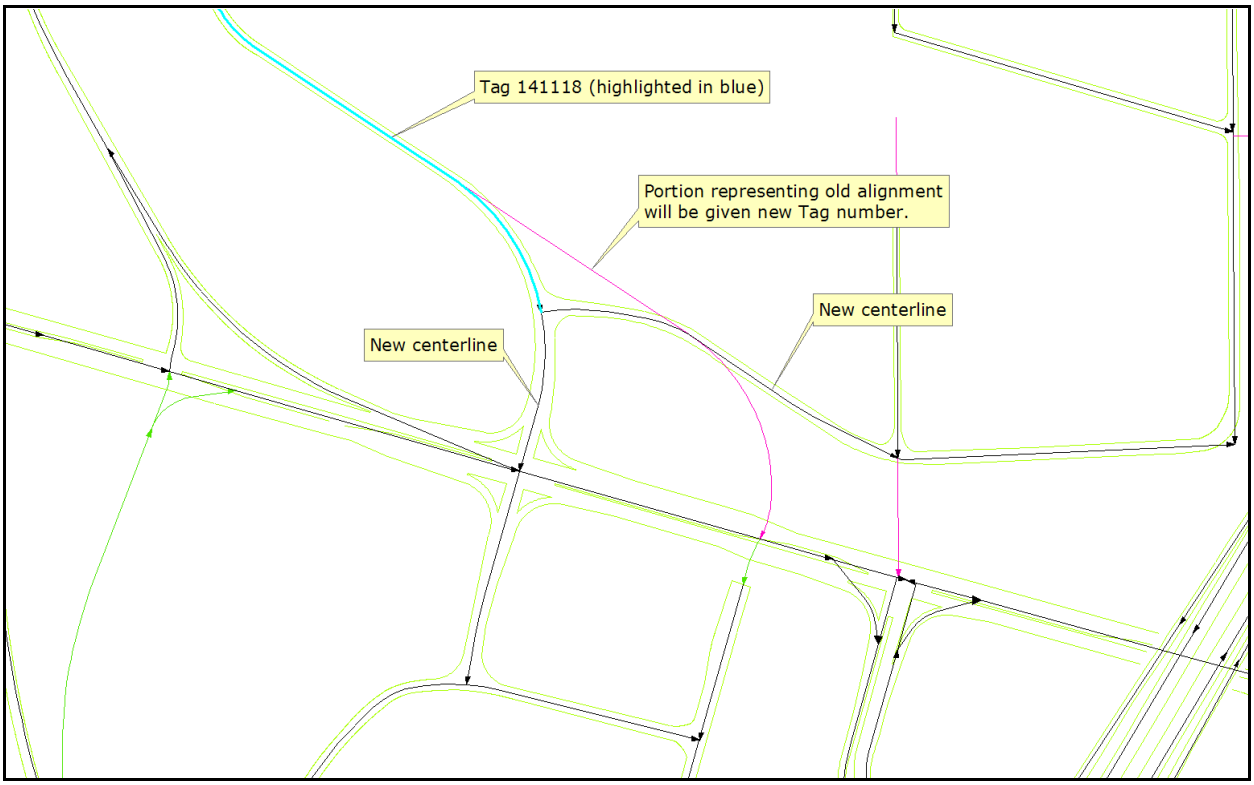


Figure 27 - Resulting centerlines after realignment.

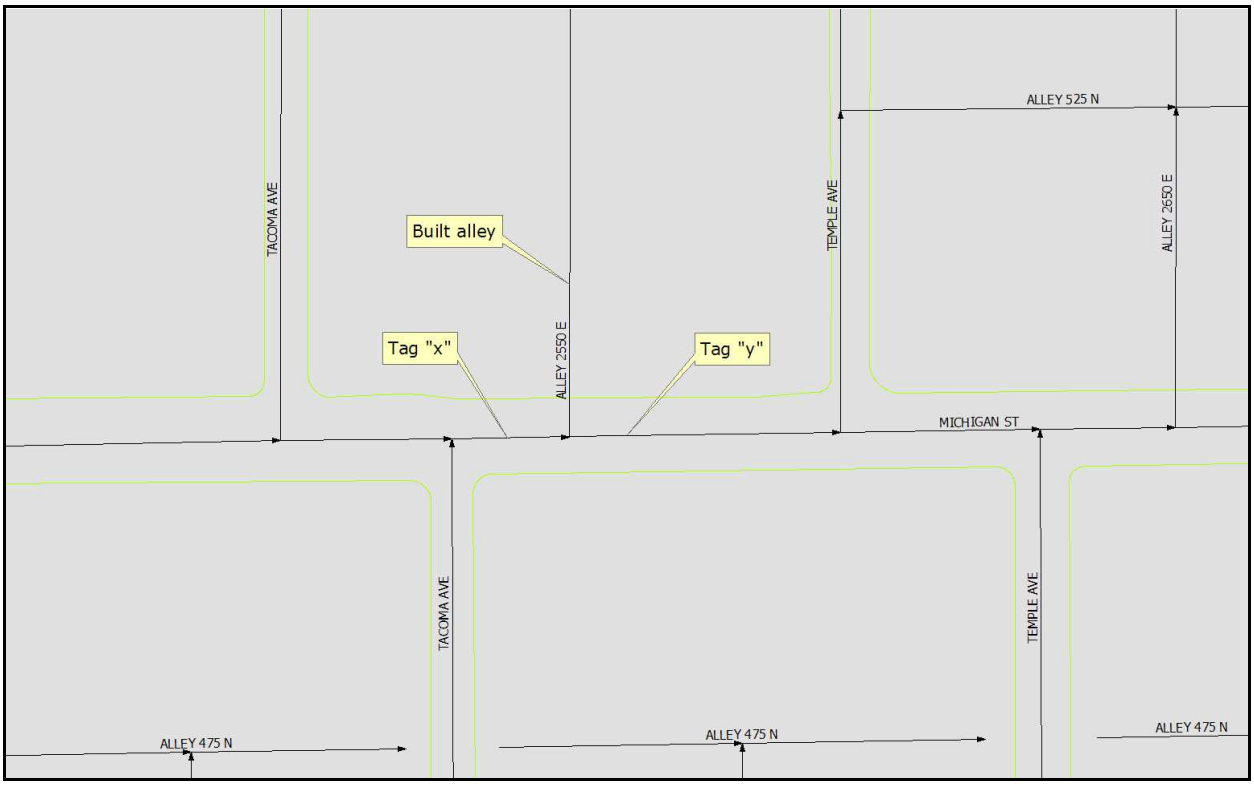


Figure 28 - Typical centerline layout before alley vacation.

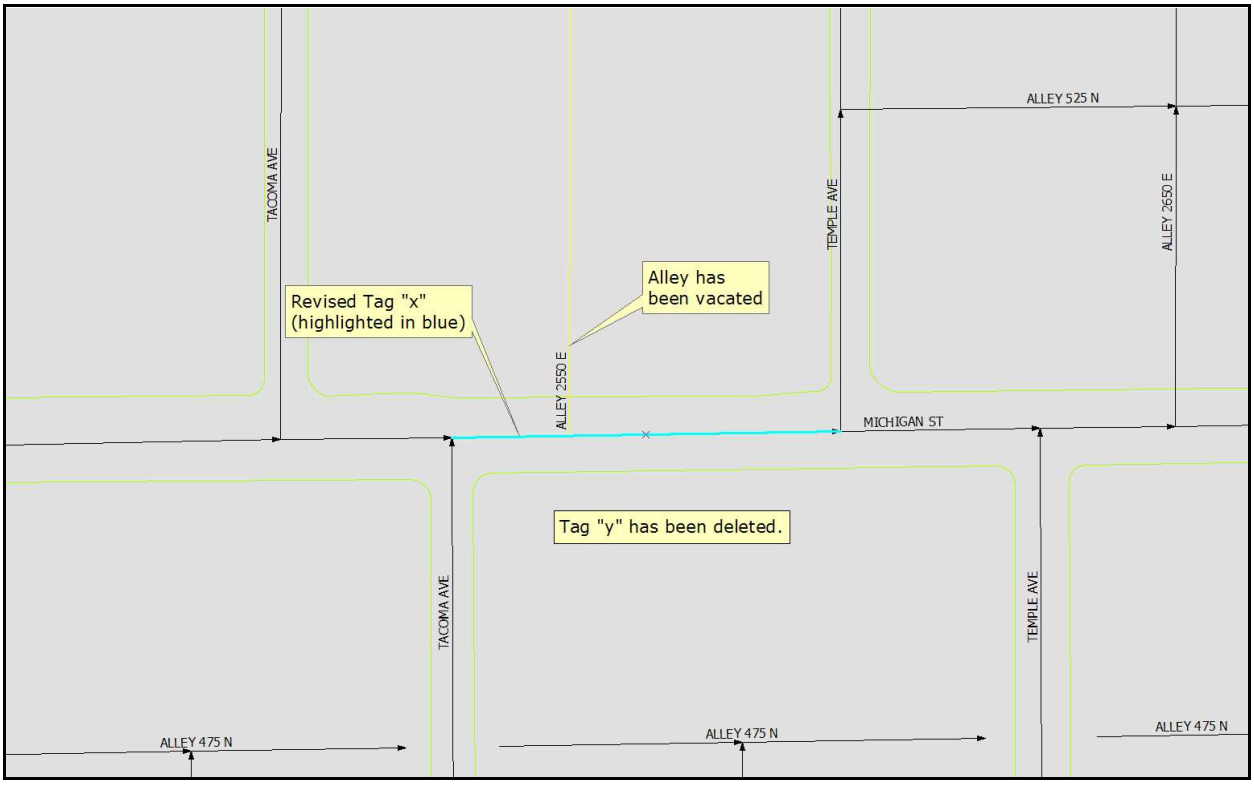


Figure 29 - Centerline layout after alley vacation.

Address Ranges

1. Address ranges will not be assigned to the following street segments:

   a. Streets in the *StreetsOOC* layer.

   b. Interstates.

   c. Streets for which there is a parallel or divergent street with the same name, which already contains address ranges. (The purpose of this rule is to prevent duplicate or overlapping ranges on the same street — required for proper geocoding.) In these situations, address ranges will normally be assigned to the segments with the higher volume of traffic. The segments without address ranges will be flagged, using the NO_ADDR attribute (see page 47). See Figures 30 and 31, next page.

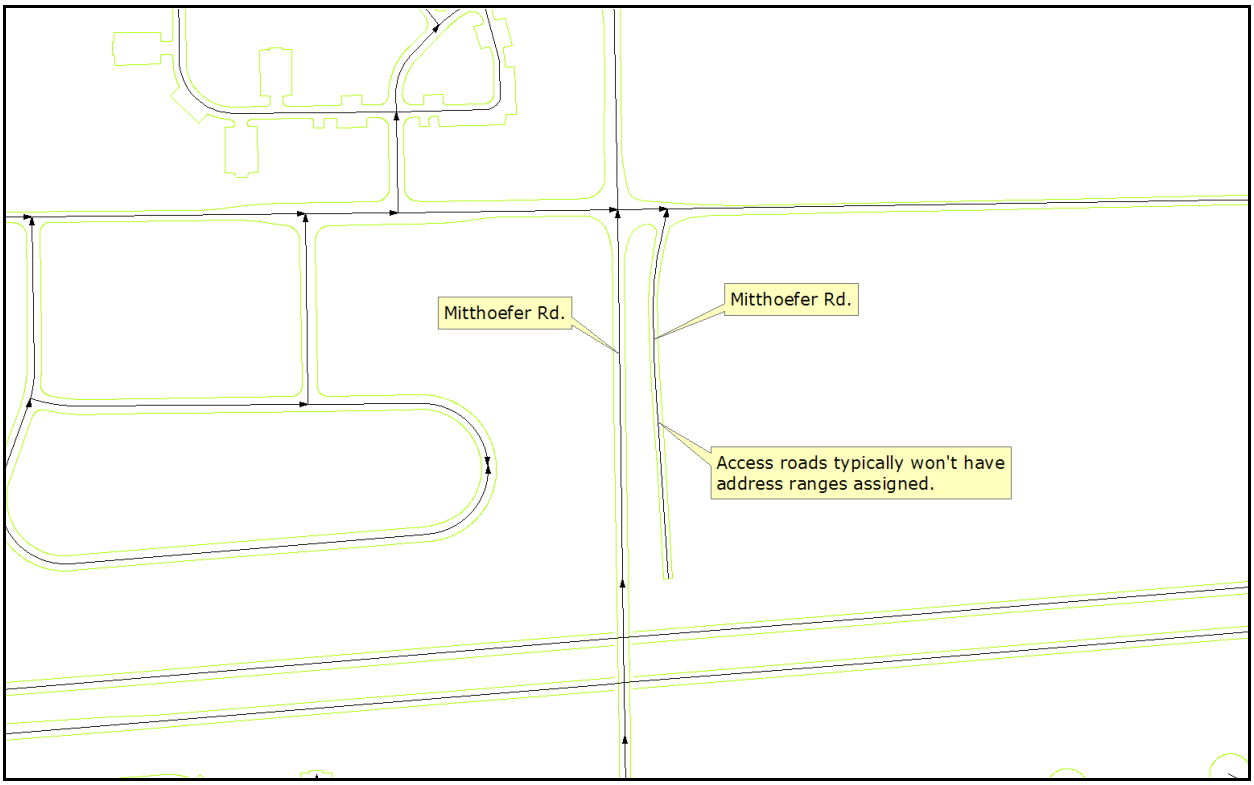


Figure 30 - Example of parallel streets with the same name.

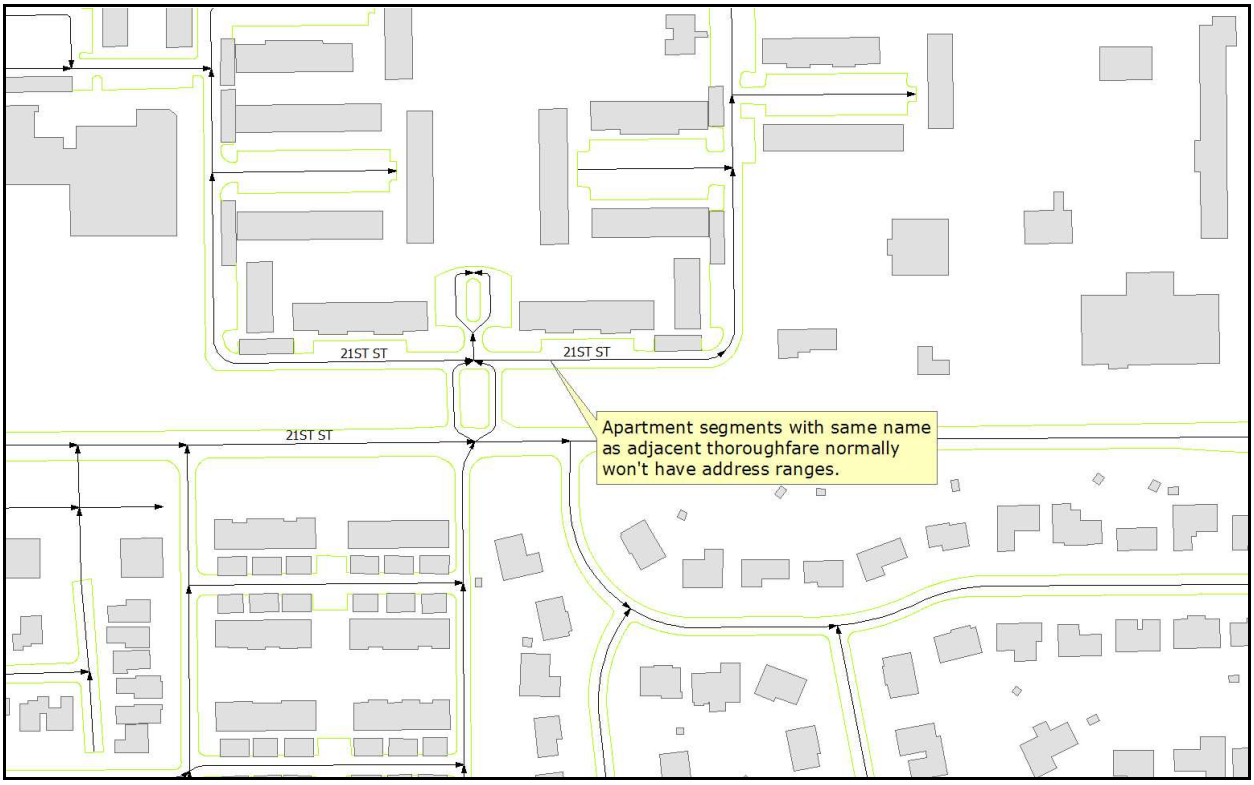


Figure 31 - Another example of parallel streets with the same name.

2. Address ranges will not be required for certain segments. Those segments will be flagged to indicate as such. They are:

   a. Segments that are too small to have meaningful addresses assigned to them.

   b. Segments for which there is no property directly addressed off of them, or for which the addresses in the area may be assigned to other centerline segments, without compromising positional accuracy. (These situations occur often in apartment and condominium complexes.) See Figure 32.

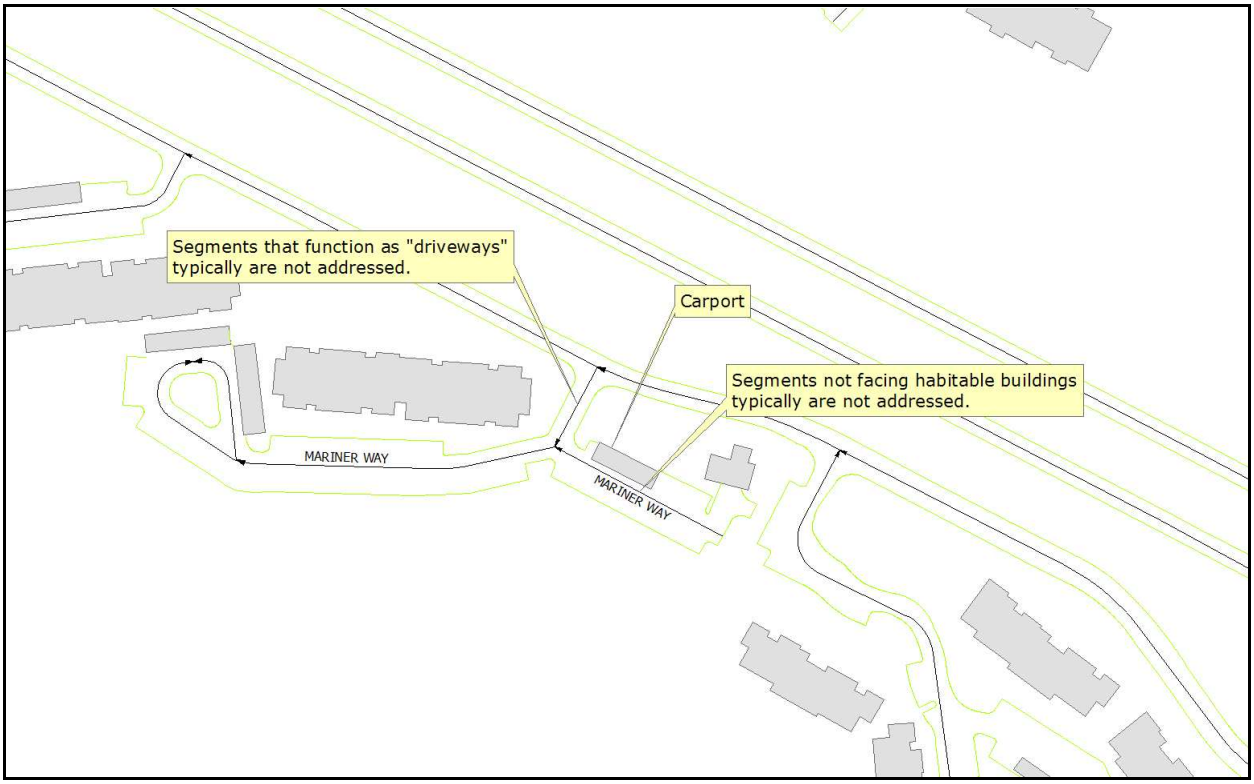


Figure 32 - Another example of non-addressed street segments.

3. All addressable centerline segments will have both left and right address ranges populated, unless it makes sense to put the even range on one segment and the odd range on another segment. There are two situations where this occurs:

   a. Wye intersections.

   b. There are two parallel pieces of pavement, with the same street name, and the addressable buildings/parcels on one side of the street are adjacent to one centerline, and the buildings/parcels on the other side of the street are adjacent to the other centerline. See figures 33 and 34, next page, for examples of these situations.

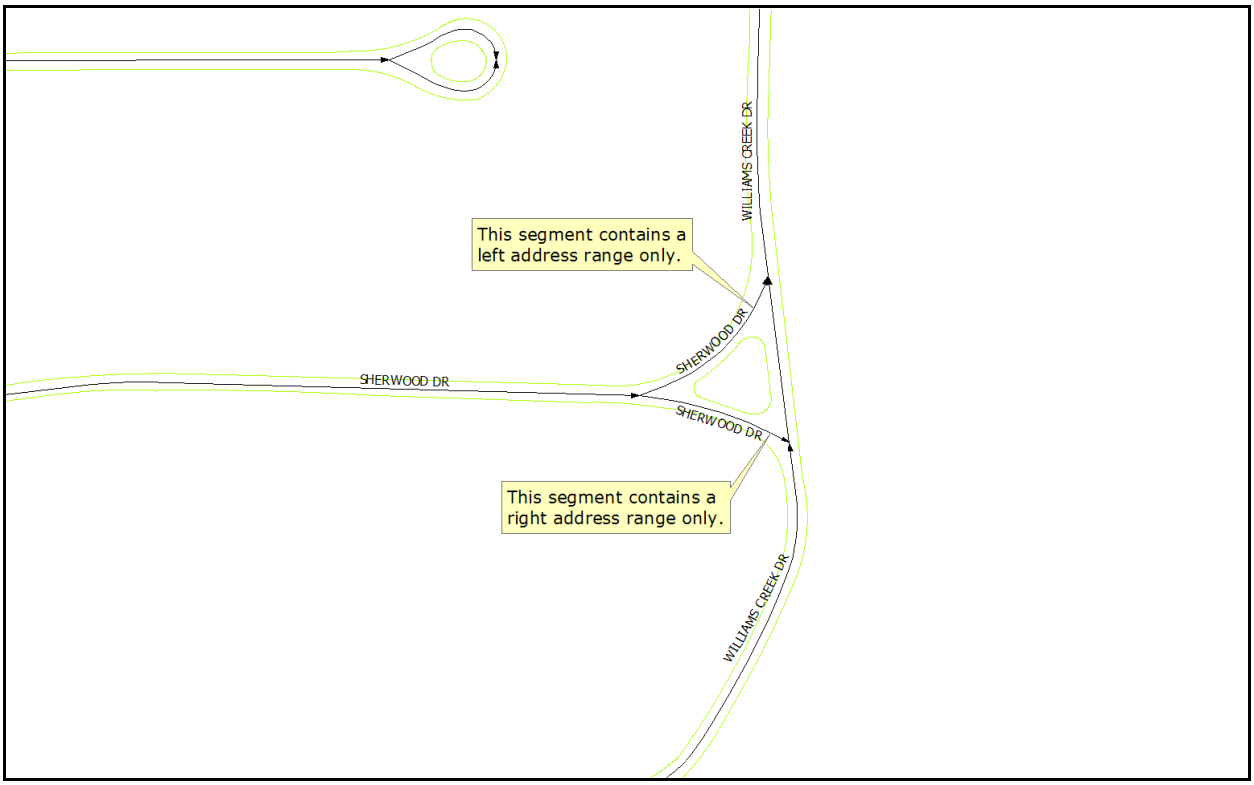


Figure 33 - Wye intersection with left and right address ranges on separate segments.

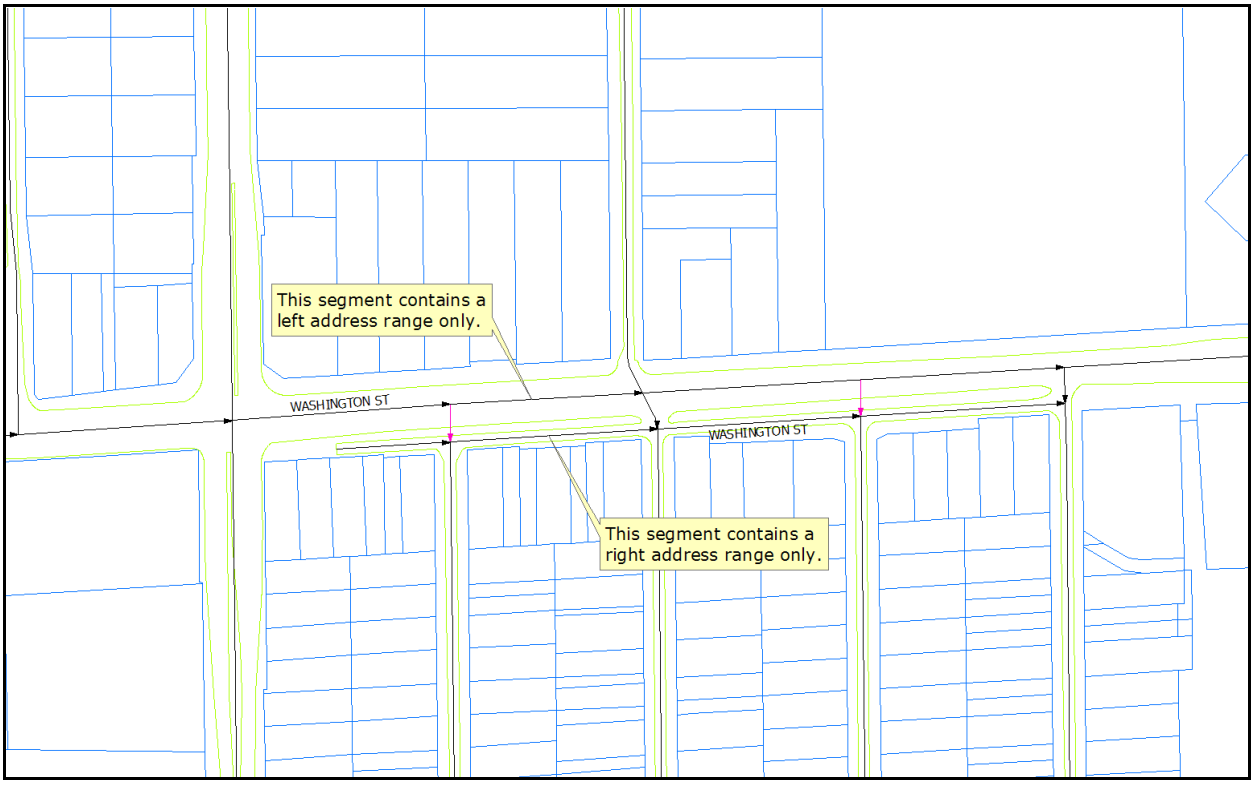


Figure 34 - Street with parallel pieces of pavement, with separate left and right address ranges.

4. I try to observe these rules, in order to assign address ranges in a prudent manner:

   a. Parcel addresses govern over building addresses, and building addresses govern over the Marion County address grid. Normally, addresses increase as you go further away from the address grid origin, but there are exceptions, especially in apartment complexes. Exceptions are noted in the *Remarks* field. (See page 50.)

   b. There should be no gaps or overlaps in address ranges along contiguous segments of the same street. If a street contains multiple parallel pieces of pavement, no portion of any address range can be duplicated on any of the pieces of pavement. (This is required for proper geocoding.)

   c. "From" addresses are always lower than or equal to "to" addresses. Again, required for proper geocoding.

   d. I try to make the lowest and highest address ranges of a street a multiple of five. (We publish a report, called the *Street Guide*, that lists all the streets in Marion County, with their beginning and ending address ranges, and I think it looks nice if as many as possible of the ranges in the report are multiples of five.) Exceptions:

      (1) The ending address range on a dead end street will be the address of the last parcel or building on the street, assuming the building or parcel is located at the end of the street.

      (2) Address ranges for segments that begin at one of the Marion County address grid baselines (Meridian St., Washington St./Rockville Rd.) will begin with "1," not "0."

      If the first or last segment of a contiguous portion of a street ends at an exact hundred block, the even ranges on the first and last segments will terminate in the exact hundred block number. For example, the westernmost segment of Sumner Avenue (located at 1500 west) will have a left-from address of 1500, and the easternmost segment (located at 3800 east) will have a left-to address of 3800 (see Figure 35 on next page).

   e. For streets that are intersected exactly at hundred blocks, the address ranges ending in "98" (even ranges) or "99" (odd ranges) are assigned to the segment entering the intersection, and the ranges beginning in "00" or "01" are assigned to the segment leaving the intersection.

   f. For individual ranges in between the lowest and highest ranges of a street, in the absence of other constraints, I assign odd addresses one number higher than their corresponding even addresses. For example, if the left-from address is 5620, then the right-from address will be 5621, not 5619 or some other number. The same rule applies to the "to" addresses.

   g. I consider how addresses on a segment will be located, if geocoded. Therefore, if a street segment contains adjacent building addresses but no parcel addresses, I might assign the lowest and/or highest ranges so that the building addresses will geocode to their actual locations more accurately. (This commonly occurs in apartment developments.) For example, consider the example in Figure 36, next page. Lake Terrace Place in an apartment complex has four buildings addressed off it. You could assign an address range to the centerline of 7615 to 7630 (or even 7600 to 7650, if you like nice round numbers) which would cover all the included addresses. Strictly speaking, even 7615 to 7630 would work. However, I would assign something like 7610 to 7635, because that way, if you geocoded the building addresses, the points would fall closer to the center of the actual buildings.

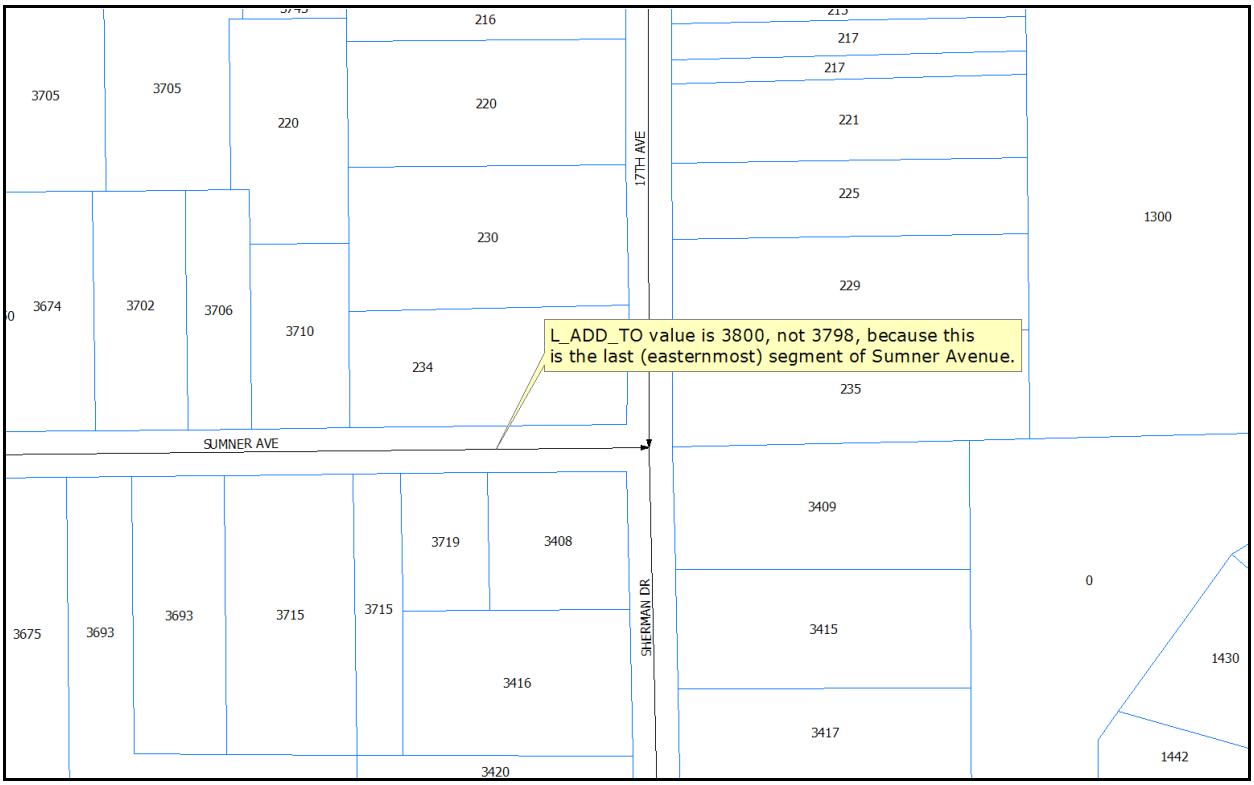


Figure 35 - Example of ending address range on the last segment of a string of contiguous segments of a street.

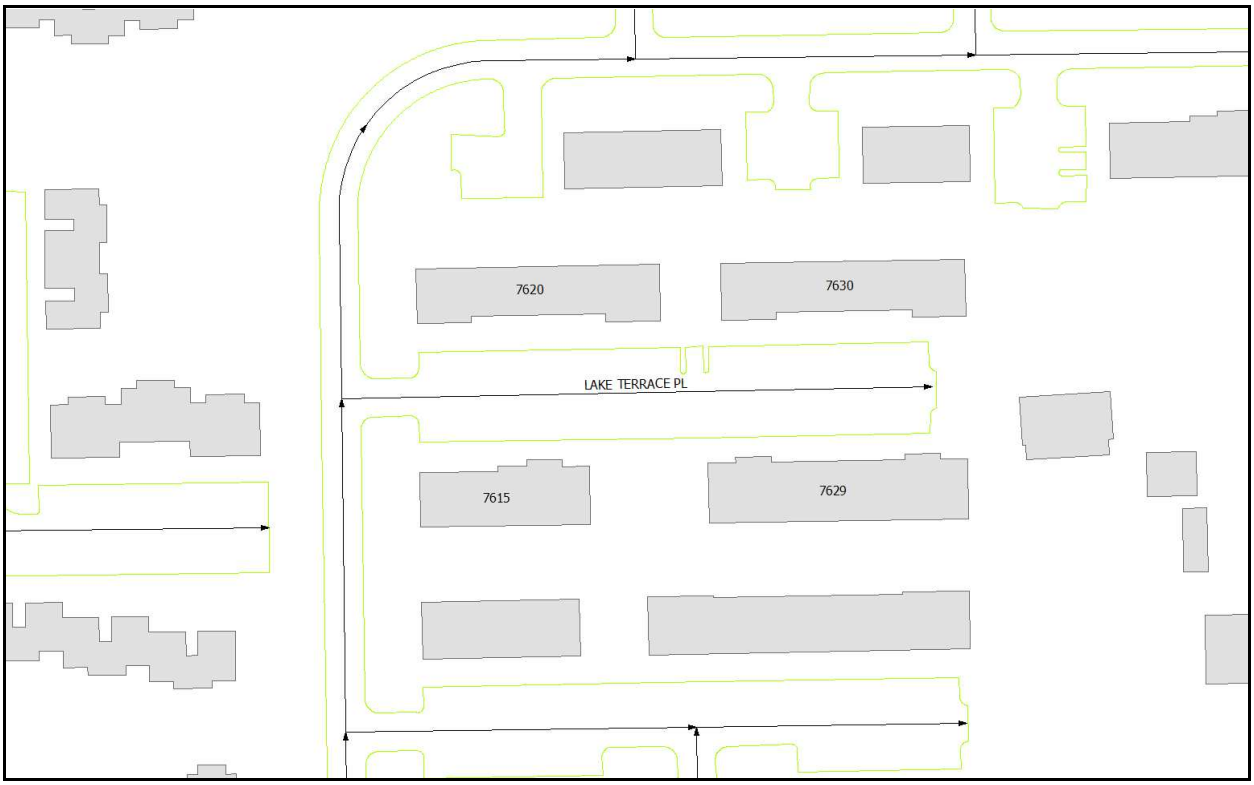


Figure 36 - Sometimes judicious assignment of address ranges can improve geocoding.

There is one exception to the above rule. If one of the ends of the segment intersects a major thoroughfare, the address of that end will not be altered, but instead will be assigned the same address as the address coordinate of the thoroughfare. For example, imagine a thoroughfare running north and south, located at 6000 east, and an apartment centerline intersecting it from the west. The "to" address of the apartment centerline would also be 6000 east. In this case, the fact that the thoroughfare lies exactly on the address grid at 6000 east overrules the desire to assign the address ranges for optimum geocoding. (See Figure 37 below for an actual example.) Since most thoroughfares in Marion County are located along the address gridlines in multiples of 1000, this policy helps in locating minor streets in the *Street Guide* that begin or end at a thoroughfare.

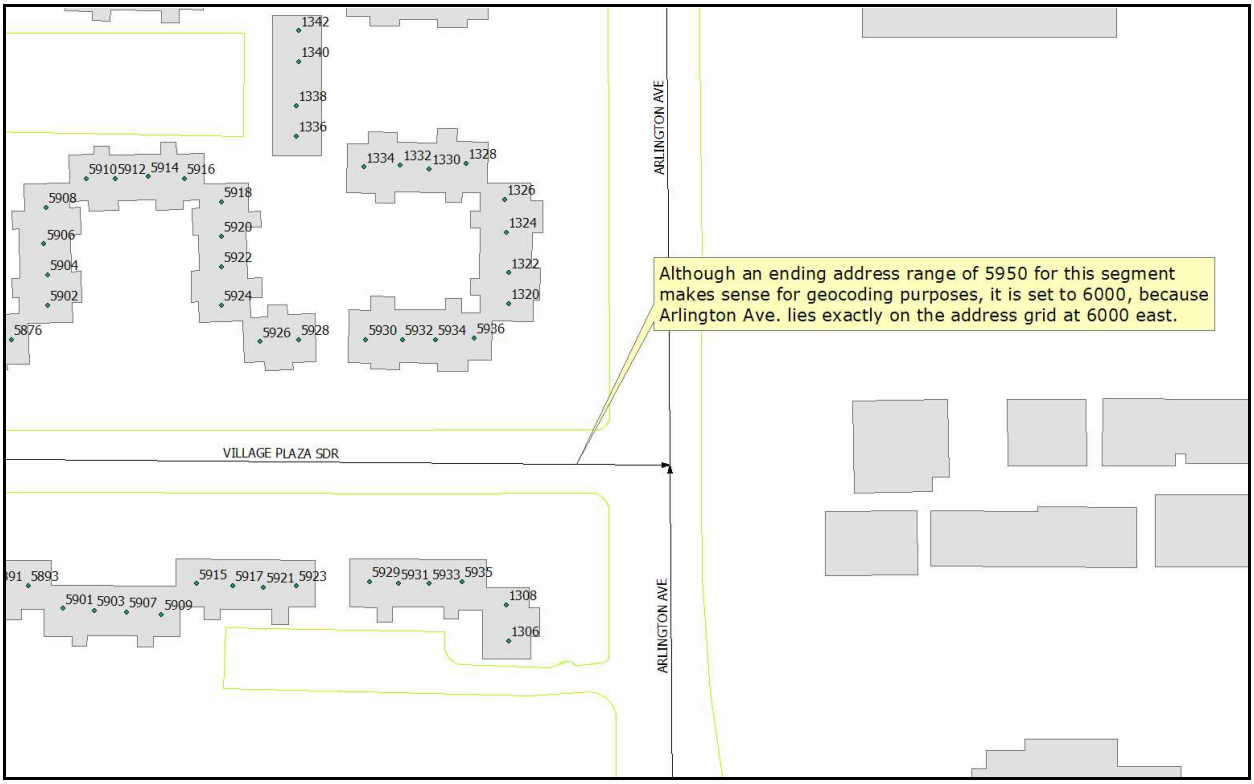


Figure 37 - Address ranges of minor streets that intersect major gridlines are governed by the address of the gridline.

MILE_FROM, MILE_TO - The beginning and ending milepost numbers of a centerline, as assigned by the Indiana Department of Transportation. Only used on interstate and state highway segments. Currently only the interstates are populated, due to a lack of data for the highways. Not populated in the *StreetsOOC* layer. These attribute values are desirable to aid in locating incidents along the interstates where the location is referenced to a milepost number. The milepost values are entered to the hundredth of a mile.

The source of the data is a milepost layer I have that I obtained from Steve Green of the Marion County Surveyor's Office. Their staff went out in the field and obtained GPS points for the mileposts in Marion County by setting up their GPS unit beside the mile marker signs along the road. I imported the "x" and "y" values to create the *Mileposts* layer. I assign milepost values to the interstate centerlines by using this layer as a reference.

To increase the resolution of the *Mileposts* layer, I added points to the *Mileposts* layer to mark the tenth-of-a-mile locations. To do this, I created a temporary centerline segment between adjacent milepost markers, and divided that line into ten equal parts. Then I placed points in the *Mileposts* layer at those endpoints. By having markers at the tenth-of-a-mile locations, I am able to interpolate the milepost values for the centerline breaks that fall between mile markers more accurately.

L_ZIP, R_ZIP - Zipcodes of the parcels on the left and right sides of the centerline. Useful for geocoding. The values are obtained from our *Zipcodes* layer. Note that the value refers to the parcel to the left or right of the centerline, not the area to the absolute left or right of the zipcode boundary. See Figure 38 below.

The zipcode domain contains values for all Marion County zipcodes, as well as all zipcodes bordering Marion County. This attribute is not totally populated in the *StreetsOOC* layer.

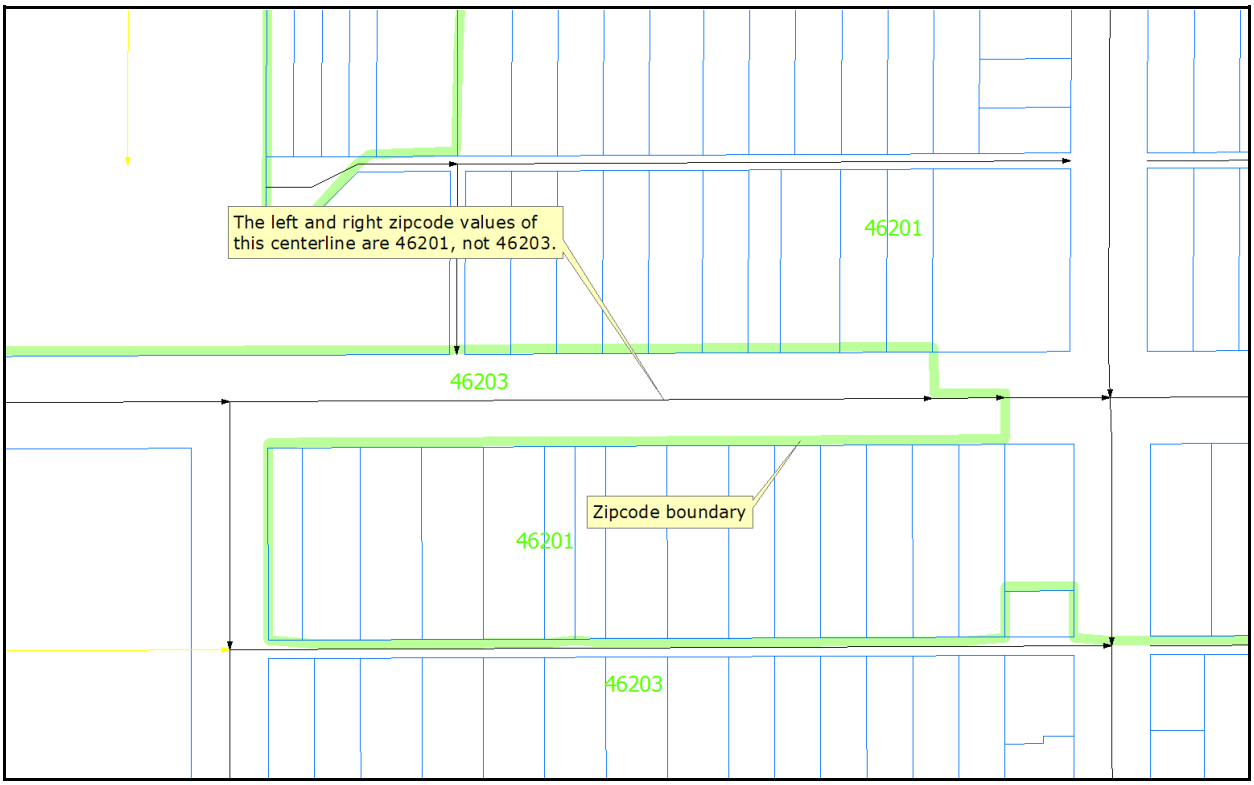


Figure 38 - Zipcode attribute values are based on the adjacent parcels, not the area to the absolute left or right of the centerline.

The domain of the zipcode fields is as follows.

46077	**46107**	**46113**	**46126**	**46140**
46142	**46143**	**46163**	**46168**	**46183**
46201	**46202**	**46203**	**46204**	**46205**
46206	**46208**	**46214**	**46216**	**46217**
46218	**46219**	**46220**	**46221**	**46222**
46224	**46225**	**46226**	**46227**	**46228**
46229	**46231**	**46234**	**46235**	**46236**
46237	**46239**	**46240**	**46241**	**46250**
46254	**46256**	**46259**	**46260**	**46268**
46278	**46282**			

L_TRACT, R_TRACT - Left and right census tract number of the centerline, based on the 2000 census. Values are from our *Census00* layer. Only populated for centerlines within or on the boundary of Marion County. If the left or right side of the centerline is outside Marion County, the corresponding attribute is left blank.

COUNTY_LEFT, COUNTY_RIGHT - County name of the parcels on the left and right sides of the centerline. The reason the attribute refers to the parcels adjacent to the centerline, and not the area immediately on either side of the centerline, is because the centerlines following county boundaries don't follow the boundaries exactly, and the purpose of the attribute is to supply the name of the county along the centerline as it would be used in common usage, not in a technical or legal sense. (See example of similar concept referring to CITY_LEFT and CITY_RIGHT attributes below.)

I obtained the values for these attributes from a layer I have locally that shows the boundaries of what is popularly called the "nine county" area (Marion County and its surrounding counties). (See page 127.) The domain follows.

MARION
BOONE
HAMILTON
HANCOCK
SHELBY
JOHNSON
MORGAN
HENDRICKS

CITY_LEFT, CITY_RIGHT - City name of the parcels on the left and right sides of the centerline. The reason the attribute refers to the parcels adjacent to the centerline and not the area immediately on either side of the centerline, is because the centerlines following city boundaries don't follow the boundaries exactly, and the purpose of the attribute is to supply the name of the city along the centerline as it would be used in common usage, not in a technical or legal sense. (See Figure 39, next page.) This attribute is only populated for centerlines within or on the boundary of Marion County. If the left or right side of the centerline is outside Marion County, the corresponding attribute is left blank.

The value of these attributes is "INDIANAPOLIS," unless the centerline lies in one of the four "excluded" cities within Marion County. ("Excluded" means those Marion County cities that were excluded from "Unigov" — see *Geographic Extent*, page 4.) These values were populated using the *Exclcity* layer. The values are:

INDIANAPOLIS
BEECH GROVE
LAWRENCE
SPEEDWAY
SOUTHPORT

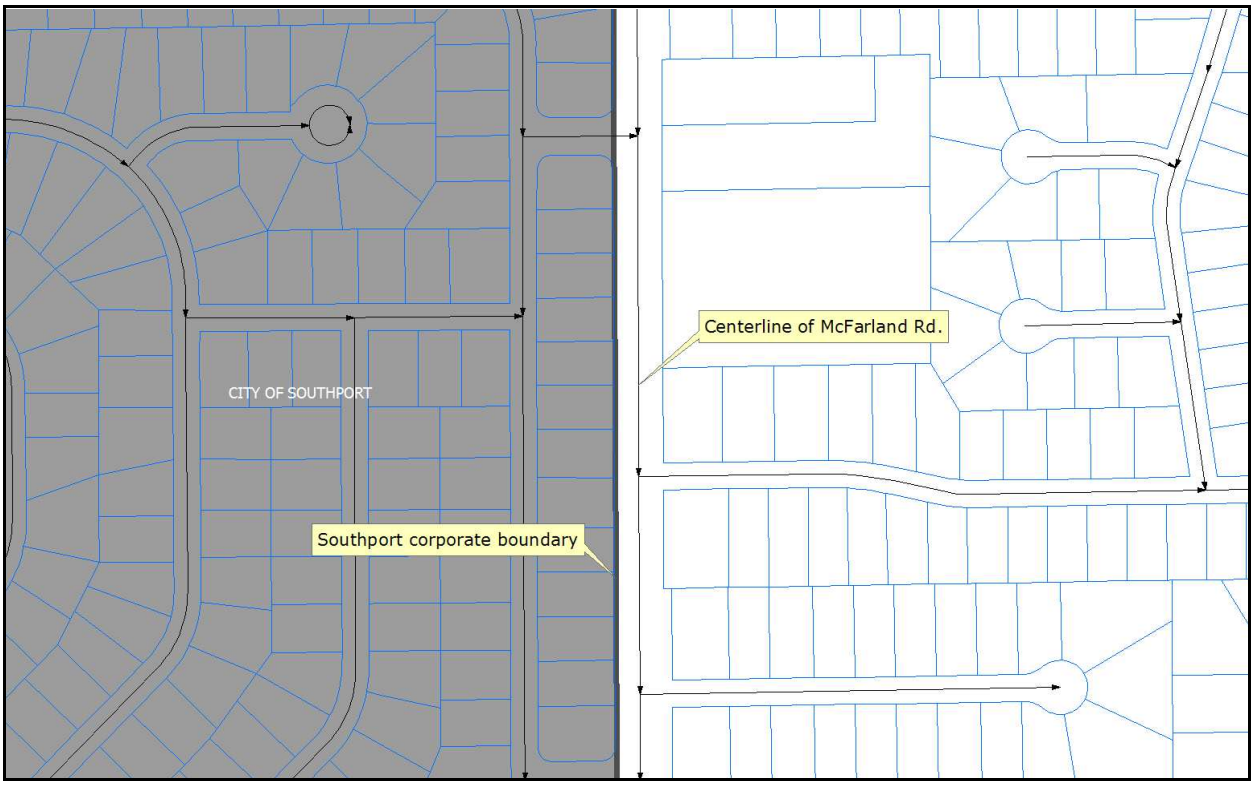


Figure 39 - The city name on the west side of the identified centerline of McFarland Rd. is listed as Southport, although technically it is Indianapolis. The gray shading denotes land within Southport's boundary.

TWP_LEFT, TWP_RIGHT - Township name of the parcels on the left and right sides of the centerline. The reason the attribute refers to the parcels adjacent to the centerline and not the land immediately on either side of the centerline, is because the centerlines following township boundaries don't follow the boundaries exactly, and the purpose of the attribute is to supply the name of the township along the centerline as it would be used in common usage, not in a technical or legal sense. (See example of similar concept referring to CITY_LEFT and CITY_RIGHT attributes above.) This attribute is currently not totally populated in the *StreetsOOC* layer.

The values in the TWP_LEFT and TWP_RIGHT attributes were populated using the *Twps* layer. The domain of the attributes is listed below. The values are listed in the same order as they appear in the picklist of the attribute editor when you click on the fields, which is the order they were entered into the geodatabase. First are the nine Marion County townships, followed by the townships that surround Marion County (mostly in alphabetical order).

CENTER
DECATUR
FRANKLIN
LAWRENCE
PERRY
PIKE
WARREN
WASHINGTON
WAYNE
BROWN
BUCK CREEK
CLARK
CLAY
DELAWARE
EAGLE

FALL CREEK
GUILFORD
LINCOLN
MADISON
MORAL
PLEASANT
VERNON
SUGAR CREEK
WHITE RIVER
FALL CREEK

OPER_STATUS - Operational status. Explanation of the domain follows.

PLANNED - The street segment is planned to be built sometime in the future, and the right-of-way has not yet been platted. All the proposed segments in DMD's *Official Thoroughfare Plan* fall into this category, as well as any other miscellaneous streets I happen to know about (such as the access roads to the new terminal at Indianapolis International Airport).

This value is also used for private centerlines that started out as "PLATTED," but for which it appears will never be built. (See explanation of PLATTED, below.) For example, sometimes a new apartment complex will be designed, and Brian Schneider will give me the centerlines for it. They will go in as platted. Eventually, for whatever reason, only part (or maybe none) of the complex gets built. When I find such a complex (from looking at the aerial photos), where it looks like all the construction is done, but there are some centerlines that were never constructed, I change the operational status of them from PLATTED to PLANNED. That way, I don't have to keep coming back to those centerlines in future years and looking at the new aerial photos to see if those streets were built yet.

PLATTED - This value has two uses. The first is for public streets, where the right-of-way has been platted, but no traversable surface exists (either because the street hasn't been built yet, or it was never built, for whatever reason). This situation most commonly occurs in new and old subdivisions and additions.

The second use for this value is for new private streets (e.g., apartments) that haven't been built yet, even though these types of streets are not technically "platted."

This value is only used for those streets within Marion County. (Platted streets outside the county are not tracked.)

In the absence of any other means of notification, the status of PLATTED streets will be changed to "BUILT" as soon as the pavement or any other evidence of substantial construction shows up on aerial photos. Conversely, in the absence of any other means of notification, the status of streets where the pavement has been removed will be changed to reflect as such, as soon as it is evident from aerial photos that the pavement is gone. See Figure 40 next page.

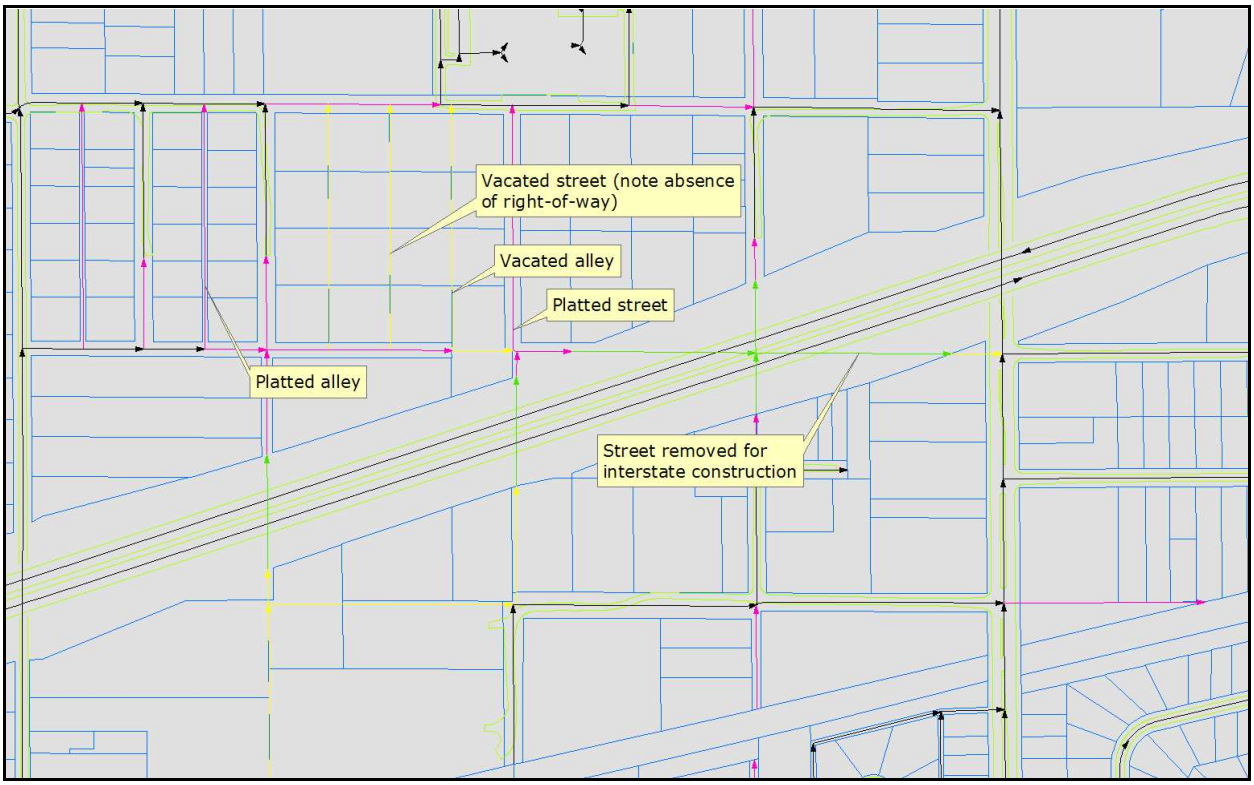


Figure 40 - Examples of platted, vacated, and removed streets.

PLATTED/LOCATOR - Same as "PLATTED," except used to denote those segments off of which a developed parcel is addressed, but from which the property owners obtain access to the parcel by way of another street, or from a driveway. This category was made necessary in order to be able to include these segments with the *Cntrlin* layer, from which geocoding is done. (See page 91.) If the segment were merely assigned the PLATTED value, by definition it would not be included in the *Cntrlin* layer, and thus would not be available to be geocoded against.

This value is used only for streets within Marion County. See Figures 41 and 42, next page.

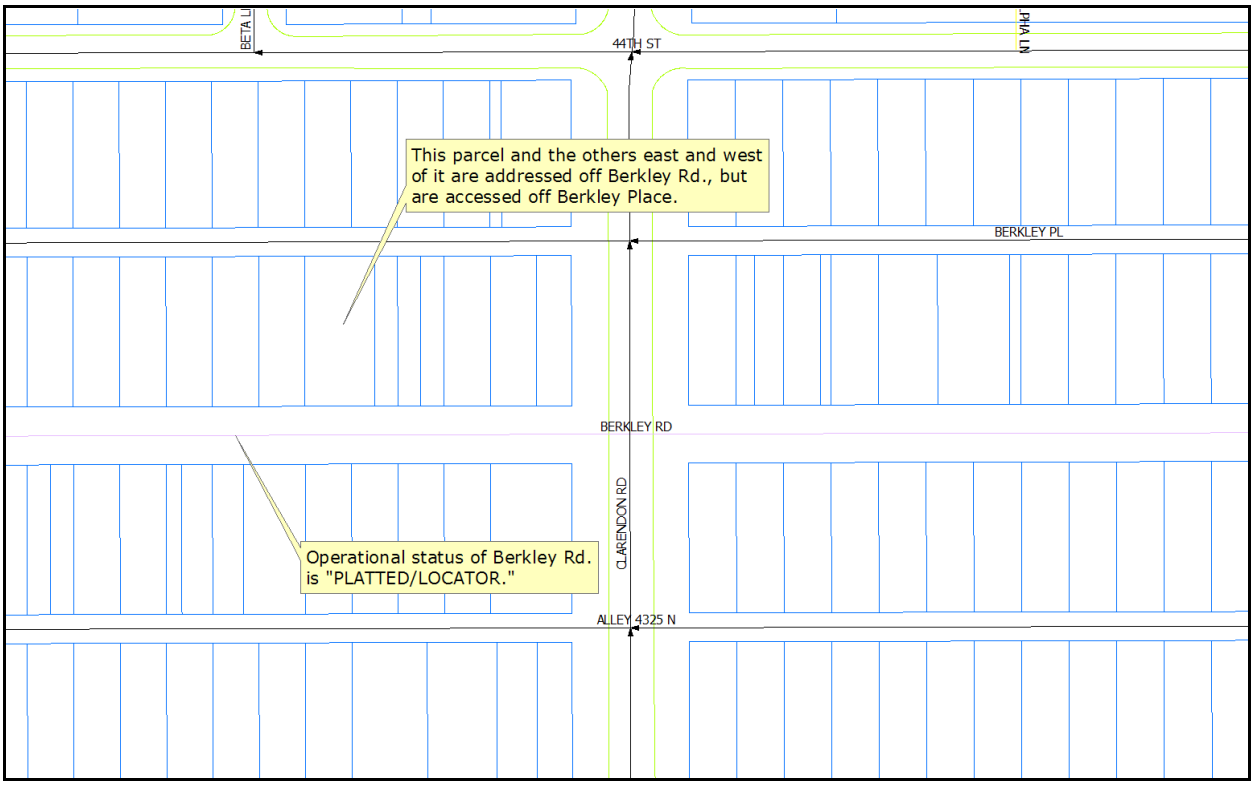


Figure 41 - Example of the use of the "PLATTED/LOCATOR" value.

Figure 42 - Another example of the use of the "PLATTED/LOCATOR" value.

BUILT - The segment is traversable. Surface can be pavement, gravel, or even dirt, but not grass or other vegetation.

VACATED - The segment has been vacated, and is not traversable. Used to indicate those segments for which I have copies of the vacation petition, or some other supporting documentation. This term also applies to those segments that I *suspect* have been vacated, based on certain conditions (absence of right-of-way, alignment of other streets in the vicinity, etc.). This value is only used for those streets within Marion County. (Vacated streets outside the county are not tracked.) See Figure 40 two pages previous for examples of vacated streets.

Streets that have become newly vacated will be changed to reflect as such, as soon as a copy of the Vacation Petition is received. Vacation Petitions are filed in my office, by DMD Basemap number, with the date the information was entered on the centerlines written at the top. Historical streets that have been vacated will also be indicated as such, regardless of whether or not the Vacation Petition number or any other supporting information is known. Supporting documentation is stored in the VACATED field (see page 61).

VACATED/LOCATOR - Same as "VACATED," except used to denote those segments off of which a developed parcel is addressed, but from which the property owners obtain access to the parcel by way of another street. This category was made necessary in order to be able to include these segments with the *Cntrlin* layer, from which geocoding is done. (See page 91.) This value is only used for those streets within Marion County.

VACATED/BUILT - Same as "VACATED," except that the segment is built (traversable). When a traversable segment is initially vacated, it will carry this designation. Later, if it is apparent from new aerial photos that the pavement has since been removed, the operational status will be changed to simply "VACATED." This value is only used for those streets within Marion County.

REMOVED - This value has two uses, depending on whether a street is public or private. For private streets, this value denotes those segments for which the pavement has been removed, as apparent from aerial photos. This is the equivalent of VACATED for public streets.

For public streets, this value is used to indicate those segments where pavement has been removed from the right-of-way, but the right-of-way still exists. This typically happens in one of two situations.

1. When the interstates were built, portions of many cross streets were removed, where they intersected the interstate alignment. The right-of-way of these portions of streets was not vacated. (Figure 40 contains examples.)

2. Sometimes a street is realigned, but the old alignment is still within the right-of-way. Figure 43 on the next page is an example.

This attribute is not populated outside Marion County.

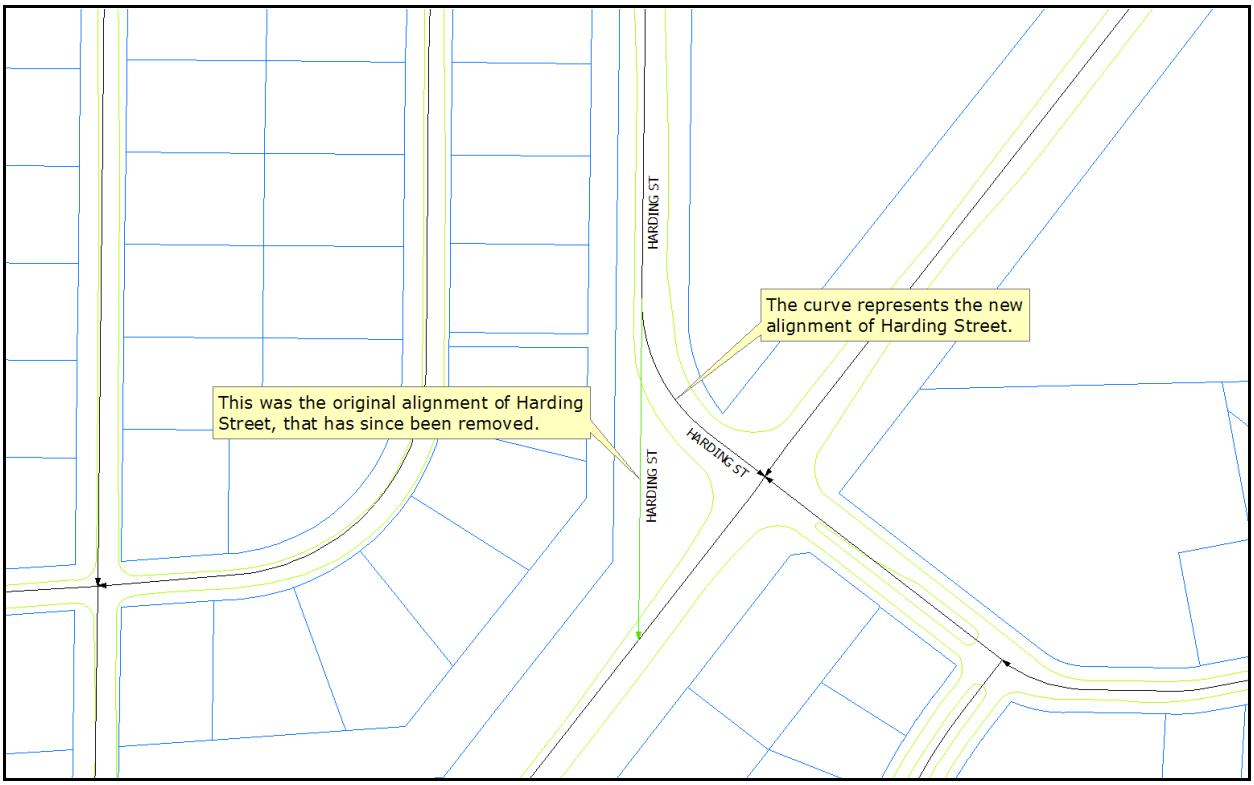


Figure 43 - Example of a street where part of the street has been removed, and the removed portion is still within the right-of-way.

Street Life Cycles

1. Public streets - Most public streets will start out as "PLATTED." The exception is "PLANNED" streets, of which there is a very small number compared to "PLATTED" streets. "PLANNED" streets become "PLATTED" streets when their right-of-way is platted. When "PLATTED" streets are improved, their operational status changes to "BUILT." If the pavement is later removed, how they are designated is determined by whether or not they are vacated. If not vacated, their status reverts to "PLATTED." If they are vacated, the status becomes "VACATED." If they are vacated but the pavement is not removed, the status becomes "VACATED/BUILT."

2. Private streets - A private street starts out as "PLATTED," then when the pavement is constructed, changes to "BUILT." If the pavement is later removed, the status changes to "REMOVED." If the pavement is never constructed, the status changes to "PLANNED."

MAINT_JURIS - Maintenance jurisdiction. Intended to show which agency has responsibility for the maintenance of the right-of-way and pavement. Explanation of the domain follows.

DPW - Indianapolis Department of Public Works. All streets within Marion County that don't fall into one of the other categories below carry this designation, as well as the streets that form the southern and eastern borders of the county (per Indiana statute).

DPW/SPEEDWAY, **DPW/BEECH GROVE**, **DPW/LAWRENCE** - Certain streets within the excluded cities of Speedway, Beech Grove, and Southport are covered under a joint agreement between the Indianapolis Department of Public Works and the respective city. These are the streets within these cities that are also a part of DMD's *Official Thoroughfare Plan*. Typically,

the City of Indianapolis is responsible for major reconstruction of these streets, and the respective cities are responsible for minor repairs.

INDOT - Indiana Department of Transportation. Interstates, ramps, highways, and access roads. Also applies to removed portions of streets crossing interstate rights-of-way.

PRIVATE - In the absence of any other information, all streets without right-of-way are labeled private, except those within the boundaries of airports, which carry their own designation (see below). In addition, all segments whose operational status contains the word "VACATED" or "REMOVED" will have a maintenance jurisdiction value of PRIVATE. For streets outside Marion County, this value is only used if the street appears to be private based on its appearance (e.g., apartment complexes).

If I happen to know that a street is private that appears to be public (or vice-versa), I try to put a short explanation in the REMARKS field (see page 50) to explain that fact. That way, if I revisit that segment, I will know that the maintenance jurisdiction field value is not an error.

AIRPORT - The segment lies within the jurisdiction of an airport, as determined from our *Airports* layer.

DEVELOPER - Assigned to those segments of public streets whose operational status is PLATTED, and which are anticipated to be built soon. Most of the time, this applies to new subdivisions, but can also apply to newly-constructed thoroughfares. This value is only used for those streets within Marion County, and specifically, only those streets not in one of the four excluded cities. (See explanation of the excluded city values below.)

When a new subdivision is constructed in Indianapolis, the new streets are inspected for compliance with construction standards. If officially approved, the new streets are accepted into the City of Indianapolis' maintenance program. I am notified of this when I receive a copy of the Completion and Compliance Affidavit packet from DPW, which lists the streets (or portions) that have been accepted. (These affidavits are filed in my office, by DMD Basemap number, with the date the information was entered on the centerlines written at the top.)

Road contractors are required to post a Maintenance Bond, and agree to be responsible for the street maintenance of their newly-constructed streets for a period of three years from the date of acceptance on the Completion and Compliance Affidavit. (The three year period is subject to change, for particular streets, if there are extenuating circumstances. If this is the case, I will put a remark in the REMARKS field to remind me.) During this three year period, the centerlines will carry a maintenance jurisdiction value of DEVELOPER. At the end of the three year period, the maintenance responsibility reverts to the City of Indianapolis, and the maintenance jurisdiction value is changed to DPW.

Every week or so I run a query to locate those streets whose three-year maintenance bond period has expired, and whose jurisdiction needs to be changed to DPW. I have this query stored as an ArcMap expression file (see page 129).

For newly-constructed thoroughfares, it is my understanding that the three year waiting period doesn't apply, so that when I observe that a new street capital improvement project is finished (as it appears in the aerial photos), I change the maintenance jurisdiction to "DPW" immediately.

SPEEDWAY, **SOUTHPORT**, **BEECH GROVE**, **LAWRENCE** - The segment is under the jurisdiction of the specified city. Generally, if a segment falls within the boundaries of one of the excluded cities, the maintenance jurisdiction is assigned the name of that city. The boundaries are identified from our *Exclcity* layer. Exceptions are those portions of streets that are a part of the *Official Thoroughfare Plan*. (See the discussion of the DPW/SPEEDWAY, DPW/BEECH GROVE, and DPW/LAWRENCE values above.)

When the operational status of a non-private street changes from PLANNED or PLATTED to show it has been built, and the street is located within the boundaries of one of the towns or excluded cities in Marion County, the maintenance jurisdiction will be changed to the name of the town

or excluded city. There may or may not be a waiting period until the responsibility officially reverts to the excluded city or town, like for Indianapolis, but since I have no way of knowing this (I get no documentation from the excluded cities), I go ahead and change the maintenance to the city's name as soon as the street is built.

WILLIAMS CREEK, **MERIDIAN HILLS**, **CROWS NEST**, **NORTH CROWS NEST**, **ROCKY RIPPLE**, **WYNNEDALE**, **CLERMONT**, **WARREN PARK**, **CUMBERLAND**, **HOMECROFT**, **HIGHWOODS** - The segment is under the jurisdiction of the specified town. Generally, if a segment falls within the boundaries of one of these towns, the maintenance jurisdiction is assigned the name of that town. The boundaries are identified from our *Towns* layer. However, portions of thoroughfares included in the *Official Thoroughfare Plan*, and lying within the boundaries of these towns, are by law DPW's responsibility, and will carry that designation.

In a few cases, there have been agreements made between these towns and Marion County, whereby the county agreed to be responsible for a street (or portion) within the boundaries of a town, and perhaps vice-versa. If I happen to be aware of these agreements (there is currently no formal mechanism whereby I am notified), I will change the maintenance jurisdiction accordingly, and put a note in the REMARKS field to explain the deviation. This type of situation may also exist between the county and the excluded cities, or even between the county and surrounding counties, but I am not personally aware of any. If I were, I would treat those situations just like the towns.

OTHER - Used for all segments outside Marion County, except those under the jurisdiction of INDOT (highways and interstates).

OLD_NAME - One or more previous names, separated by commas, that the street segment was officially known by in the past. If a value exists in this field, it generally implies that the street was renamed by some kind of official mechanism, such as by Resolution. However, there are cases where streets have been referred to universally by a different name for so long, through common usage, that the name is considered (by me and/or the majority) to have been changed, although there may be no "official" legal record of the change. I have discovered in my years of experience with street names that sometimes there is no "official" name of a street, especially for older streets. This may be because the "official" records have been lost or are too hard to locate, or maybe the street was never "officially" named in the first place! I have learned that I need to be flexible when it comes to street names.

Note that old street names are not considered aliases, for the purposes here. (See the explanation of the ALIAS_FULL_STNAME field, page 69.)

All entries in the OLD_NAME field will follow the full street name format, just like the FULL_STNAME field (see page 68).

NO_ADDR - This field is used by myself to track centerlines that don't contain one or both address ranges (values are zero). Some NO_ADDR values indicate that the absence of an address range constitutes a problem, and some values indicate that the absence is acceptable. Domain follows.

Values that indicate the absence of an address range is a problem

APT - Either the left, or right, or both address ranges are missing, and the centerline is part of an apartment or condominium complex, or something similar.

COM - Either the left, or right, or both address ranges are missing, and the centerline is part of a commercial development, or something similar.

SUB - Either the left, or right, or both address ranges are missing, and the centerline is part of a subdivision, metes and bounds area, or some other location where there are adjacent individually addressed parcels.

Values that indicate the absence of an address range is acceptable

DIV - The centerline has a corresponding (usually parallel) centerline with the same street name and address range, that contains the missing address range. There may be more than two parallel centerlines, and the centerlines don't intersect. Since only one can contain the left and/or right address ranges, all others will be marked with "DIV." See Figures 30, 31, and 34.

WYE - The centerline has a corresponding, diverging centerline with the same street name and address range, that contains the missing address range. The two centerlines intersect. The even address range will be on one centerline, and the odd range will be on the other. See Figure 33.

INT - The centerline is too short to assign a meaningful address range to it. In the beginning, before the concept of logical intersections was adopted, there were many "offset" intersections, where the nodes of the opposing streets were separated by a very short segment. These segments were identified with the "INT" value, which stands for "intersection." Although almost all the offset intersections have been located and replaced by logical intersections, there still remain some cases where a very short segment is still appropriate. See Figure 44, next page, for an example.

NOA - The centerline doesn't have any parcels or buildings addressed off it. This typically happens a lot in apartment complexes. While this condition also meets the criteria for the "INT" value above, the "NOA" value is reserved for those cases where the "INT" value doesn't apply. See Figure 32.

CUL1 - The centerline is one side of a cul-de-sac loop, containing either the left or right address range. The other centerline making up the other side of the loop contains the opposite address range. See Figure 21.

CUL2 - The centerline represents a cul-de-sac, contains only the left or right address range, and there is another coincident, identical centerline running in the opposite direction, that contains the opposite address range. This configuration allows for more accurate geocoding in cases where properties are addressed in a circular fashion around a cul-de-sac, instead of being addressed in the same direction on both sides of the cul-de-sac. See Figure 23.

MIXED - The centerline is part of a complex network of segments that make up a single street, and the parcels on the street are addressed in such a way that it is impossible to assign the address ranges to all the centerlines in a logical, consecutive order. In this case, certain segments are selected to contain the address ranges, and the ones that are left are tagged with this value. See Figure 45, next page.

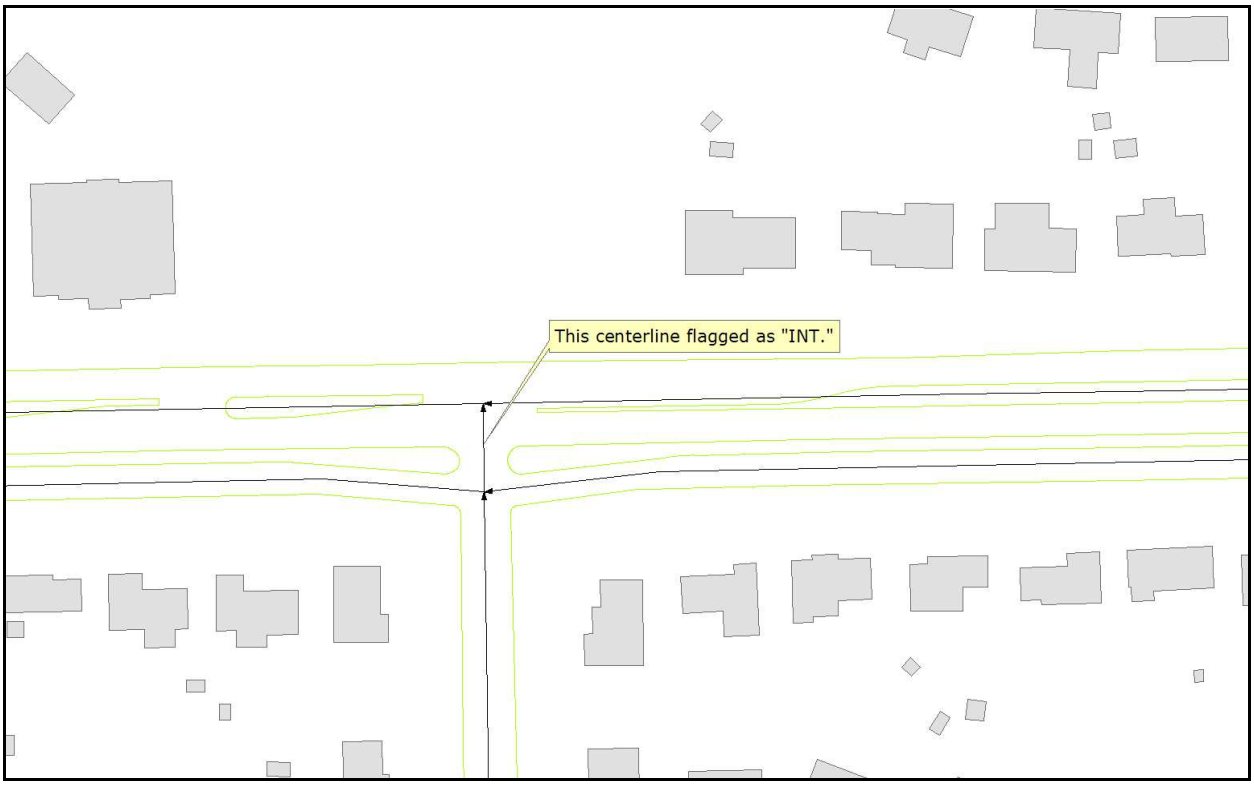


Figure 44 - Example of the use of the "INT" NO_ADDR value.

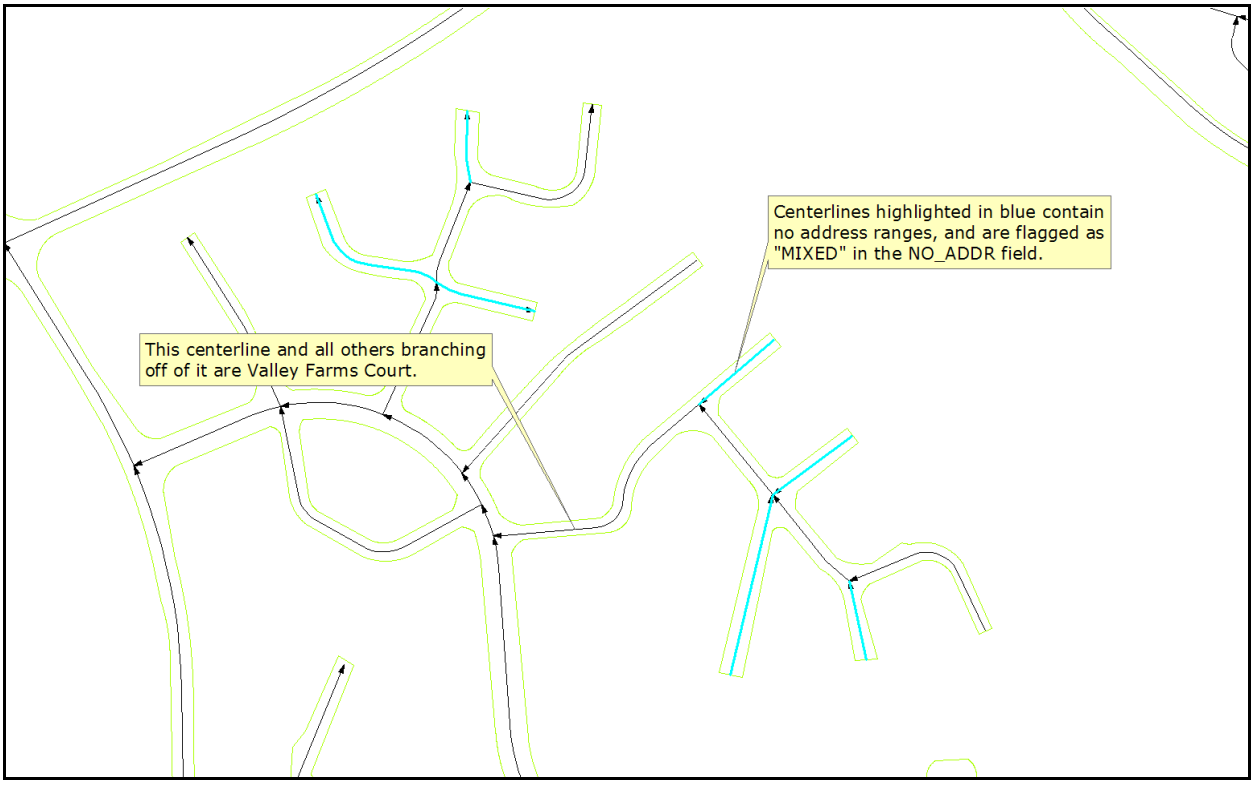


Figure 45 - Example of the use of the "MIXED" NO_ADDR value.

ADDPROB - This field is used by myself to track centerlines that contain questionable or missing values for the street name or address ranges. The value in this field will always consist of three alphanumeric characters. Each character carries a certain meaning. The first character pertains to the street name, and the second character pertains to the address ranges. The third character is useful when a check of the address ranges is made using the ArcInfo ADDRESSERRORS command.

The ADDRESSERRORS command checks for the presence of broken chains in the address ranges of streets. There are occasions where broken chains are acceptable, due to the anomalies in the way parcels in Indianapolis are sometimes addressed. This value allows one to exclude the "anomalies" before running the ADDRESSERRORS command, so that they will not be identified as errors in the output.

(Update: Since the initial time we ran the ADDRESSERRORS command against the centerlines several years ago, it has never been run again, and I have quit maintaining this portion of the ADDPROB values.)

Here is the domain of the first character of the PROB field:

0 - The street name is missing (unknown).
1 - The street name is questionable.
9 - The street name has been verified, or it is reasonably sure that it is correct.

Here is the domain of the second character of the PROB field:

0 - The address ranges are missing (unknown).

1 - The address ranges are questionable. This often happens due to the lack of adjacent parcels that could be used to verify the ranges (e.g., in apartment complexes).

9 - The address ranges have been verified, or it is reasonably sure that they are correct. Verification is by way of adjacent parcels or some other type of documentation.

Here is the domain of the third character of the PROB field:

Y - The address ranges constitute an anomaly as far as the ADDRESSERRORS command is concerned (they are part of a broken chain).

N - The address ranges do not constitute an anomaly as far as the ADDRESSERRORS command is concerned (they are part of an unbroken chain).

REMARKS - This field is kind of a "catch-all" for any information that doesn't fit in the other fields. Information that may appear in this field includes, but is not limited to, jurisdiction relinquishments, double centerline notations, abnormal bond expiration dates, and documentation supporting any of the other attribute values that may appear to be incorrect, at first glance. There can be multiple remarks for one centerline; if so, they will be separated by semicolons.

There are some standard entries that I put in this field. These remarks are always worded exactly the same, so that I can search on them easily if I want. The standard entries can be combined with other remarks, if need be.

Three standard remarks pertain to address range anomalies. These remarks are entered as a reminder to me if I ever revisit that centerline, that the address ranges were not entered in error.

Standard remarks pertaining to address range anomalies

ADDRESSED BACKWARDS - The "from" and "to" addresses are contrary to the Indianapolis address grid system. For example, a north-south segment north of Washington Street might have increasing addresses in a southerly direction, instead of a northerly direction. These situations hardly ever happen on public streets, but they happen quite frequently in apartment and condominium complexes.

REVERSE ADDRESS PARITY - In this situation, the odd and even address ranges are opposite of the standard. For example, a north-south segment might have even ranges on the east side of the street and odd ranges on the west side. Again, this situation hardly ever happens on public streets, but happens quite frequently in apartment and condominium complexes.

NONSTANDARD ADDRESS PARITY - This remark is used for all other types of address range anomalies, such as a combination of the above two situations.

Standard remarks pertaining to miscellaneous situations

DOUBLE CENTERLINE - The centerline is one of a pair of identical, coincident centerlines. This remark is used to locate overlapping centerlines, since you can't find them just by looking at them. If this remark is combined with other remarks, then this remark will always appear first. (See page 23.)

REMOVED *XXXX* - This remark is placed on centerlines for which the operational status is REMOVED, and the year the pavement was removed is known. (The *XXXX* is replaced with the removal year.)

BOND EXPIRATION *XX/XX/XXXX* - This remark is used to indicate those centerlines for which the maintenance bond expiration date is something other than the standard, which is three years after the acceptance date. See the discussion on the DATE_ACCEPTED attribute, page 70. (The *XX/XX/XXXX* is replaced with the actual expiration date.)

ENDING RANGE REQUESTED BY B. SCHNEIDER - Sometimes Brian Schneider will contact me and instruct me to change the address range on a centerline to accommodate a new address that he wants to add to the parcels. (Parcel addresses can't be added unless they already exist on the centerlines — see the discussion of the Polyline layers on page 126.) Since the centerline address ranges are based on the adjoining parcel addresses, and the new address doesn't exist in the parcels yet, without this comment in the REMARKS field, I might think the centerline address ranges are in error.

PRIVATE PER S. POWELL (DOCUMENTATION ON FILE) - Sherry Powell in the Department of Public Works is the custodian of our Pavement Management System. Consequently, she deals with street issues a lot. Sometimes, she will be privy to information that I don't know about that affects the centerlines. An example would be a street that is private, even though our *Parcels* layer is showing right-of-way for the street. In a case like this, Sherry will notify me of the discrepancy, and send me a copy of any supporting documentation that she has. I place this comment in the REMARKS field, and file the documentation for future reference in a file cabinet I have. (See *Filing System*, page 132.)

I may receive information on streets from other persons beside Sherry, which of course would change the name in the remark. Regardless, I always follow the same format when placing the comment in the REMARKS field. I also always try to get a copy of the supporting documentation for reference, in case there is any question in the future.

DATE_CREATED - The date the segment was created. The format is *XX/XX/XXXX* (month, day, year). If the number of the day or month is less than ten, the leading zero is omitted. For the purposes of this discussion, a centerline may be "created" in one of two ways.

The first way is when a new centerline is first digitized and all the attributes populated. Sometimes I digitize new centerlines one day, and populate the attributes the next day, or even several days later. In these cases, the centerline is not considered created until the last attribute is populated. That is when the new centerline is committed to the Master Address Database. That date is the one that will be applied to this attribute.

The other way a centerline may be "created" is when an existing centerline is split. In this case, one of the resulting centerlines is considered to have been "moved" (see DATE_MOVED below), and the other one "created." If the original centerline had address ranges, I consider the one with the lower resulting address ranges to be the one that was moved, and the one with the higher ranges to be the one that was created.

On the date that a centerline is split, the DATE_MOVED attribute is populated with that date for the resulting centerline that is considered moved. For the centerline that is considered created, the DATE_CREATED attribute is populated with the current date, and if the DATE_MOVED and DATE_CHANGED attributes contain values, those values are nulled out. This centerline is also assigned a new CENTERLINE_TAG value. (See CENTERLINE_TAG attribute, page 28.)

DATE_MOVED - The date the segment was last moved. The format is *XX/XX/XXXX* (month, day, year). If the number of the day or month is less than ten, the leading zero is omitted. For the purposes of this discussion, a centerline is considered to have been moved if any one of the following conditions is met:

- The entire centerline is moved to another location.
- One or more of the vertices or endpoints is moved to another location.
- One or more vertices is added to the centerline, or one or more vertices is deleted.

A centerline, by definition, can never have a DATE_MOVED or DATE_CHANGED date equal to a DATE_CREATED date. If a new centerline is moved or changed after being initially created and committed to the Master Address Database, but still within the same date it was first created, then neither the DATE_MOVED field nor the DATE_CHANGED field is updated. In other words, any changes to a centerline on the date it is created are not considered changes, as far as the DATE_MOVED and DATE_CHANGED fields are concerned. At least one calendar day must elapse since the date the centerline is created before these two fields may contain values.

DATE_CHANGED - This attribute is updated with the current date whenever any value of any attribute is changed, except for the NO_ADDR, ADDPROB, DATE_CREATED, and DATE_MOVED fields. The format is XX/XX/XXXX (month, day, year). If the number of the day or month is less than ten, the leading zero is omitted.

COORDINATE - This field was added at the request of our emergency response people. In conjunction with the COORDDIR (coordinate direction) field, it is used to help locate a particular centerline, according to where it lines up with the Marion County address grid. This allows people who aren't familiar with the particular street to instantly get a rough estimate of where the segment is located.

If a segment is addressed east-west, the value in this field will be a north-south address coordinate, and vice-versa. The value of the COORDDIR field will be either "N," "S," "E," or "W," standing for "north," "south," "east," and "west," respectively. So, for example, if a segment is addressed east-west, and it lies approximately 4800 north, then the value of the COORDINATE field would be "4800," and the value of the COORDDIR field would be "N."

Here are the policies we decided to follow when populating the COORDINATE field:

1. The value of this field will generally be rounded to a multiple of 25.

2. If a segment turns, the coordinate will be based on the longest or most predominant portion of

the centerline.

3. Segments that don't require address ranges will be excluded (includes interstates).

4. Meridian Street, which forms the north-south axis of Marion County's address grid, will be assigned the coordinate zero west ("0 W"). Washington Street and Rockville Road, which form the east-west axis of the address grid, will be assigned the coordinate zero north ("0 N").

5. The value of the COORDDIR field will always be 90 degrees to the direction the segment is addressed. Since streets running diagonally must be addressed either east-west or north-south, it is usually obvious which way the COORDDIR field must be populated.

   Note: Indianapolis has four main streets that radiate out from the center of the city at an angle of 45 degrees: Indiana Ave., Kentucky Ave., Virginia Ave., and Massachusetts Avenue. I was able to learn from an old reference (*Cram's 1970 Indianapolis--Marion County Street Guide*) that these streets are numbered as north-south streets, except that Massachusetts is numbered as an east-west street beyond ten hundred.

6. For streets within the four "excluded cities," we will assign the coordinate using the Marion County grid, even if the cities have their own address grid.

7. Streets in the *StreetsOOC* layer are excluded.

8. For streets within the four "excluded cities," we will assign the coordinate using the Marion County grid, even if the cities have their own address grid.

9. Streets in the *StreetsOOC* layer are excluded.

10. For parallel streets with the same name, assign slightly different coordinates (difference of 25) to distinguish them from each other. Normally this occurs in private development areas. This policy does not apply to streets represented by multiple centerlines separated by a median.

11. Since alleys are named according to their address coordinate (see page 66), their address coordinate will be the same one that is contained within their name.

In order to assign a value to the COORDINATE field, I look at the addresses of the other parcels in the vicinity, if there are any. Since the address grid is not uniform, but can vary significantly, I only look at the addresses in the near vicinity when determining the address coordinate. If there are no parcels (of sufficiently small size) to get a good idea of the coordinate, I look at the address ranges on the nearest centerlines of other streets. Also, knowing what the coordinates are of the major thoroughfares throughout the county helps.

This field, and the COORDDIR field, are used to create our Street Guide Report, which lists every street in Marion County, and shows at what address it begins and ends, and where on the grid it lies.

COORDDIR - See the field above for explanation.

TFARE - This field identifies the segments comprising the *Official Thoroughfare Plan* of Marion County. The domain consists of the classifications of streets, as defined by the *Thoroughfare Plan*, with the exception that Collectors and Local Streets are combined. This is because the *Thoroughfare Plan* doesn't provide any documentation on which streets are classified as Collectors.

The information to populate this field came from a printed copy of the *Official Thoroughfare Plan* (including a couple revisions). I try to get all updates of this document, so that I can keep this field current. Domain follows.

FREEWAY
EXPRESSWAY
PRIMARY ARTERIAL

SECONDARY ARTERIAL
SECONDARY ARTERIAL (2 LANE)
COLLECTOR/LOCAL STREET

STRCLASS - This field is used to classify streets, similar to the TFARE field. (This attribute used to be called "MAJOR.") Several years ago it was decided that the way the streets are classified by the *Official Thoroughfare Plan* does not lend itself the best to producing maps that show the classifications of streets, the way that most people are used to thinking about them. Therefore it was decided to create this field, so that the streets could be classified in such a way that symbolizing the different levels appropriately makes for a nice-looking map. This classification is purely arbitrary. It was created by myself, with some feedback from a few people on staff. It is close to the *Official Thoroughfare Plan*, but has minor differences. It tends to include more streets as major streets than the *Thoroughfare Plan*. Some of these streets were added to help fill in "holes" in the county; areas where major streets are widely spaced.

This field is also used to extract various subsets of the master *Streets* layer, in order to create other popular centerline-derived layers, such as *Major Streets*, *Interstates*, and *Paper Streets*. The domain follows.

A - Interstates (not including ramps). Extracting all centerlines with this value produces our *Interstates* layer.

B - Highways (present and former, but not including ramps). This value identifies those segments that make up the state highways, including their former alignments that ran through Indianapolis before the interstates were built. Extracting all the centerlines with this value, as well as the one above, produces our *Highways* layer.

C - All other major streets, but not including ramps. Extracting the centerlines with this value, as well as the ones above, yields our *Major Streets* layer.

D - All streets that don't match any of the other five criteria. Originally this was the last of the classifications, and is intended to signify minor streets.

E - Alleys. Extracting all the centerlines with this value yields our *Alleys* layer. (See *Alleys*, page 66.)

F - Ramps. Includes interstate and highway ramps. Ramps have been given names by INDOT, which I have put on the centerlines. (See RAMPS, page 63.)

STRLEVEL - This field was created so we could implement a modified form of the ESRI 1:24,000 scale Base Map Data Model. It allows us to make maps that depict overpasses with streets above other streets. Basically, all segments were initially assigned a value of "0" (zero), and then the ones that constitute overpasses, where the street passes over another street below, were assigned a value of "1." (In a few cases, where complicated interchanges exist, it was necessary to assign a value of "2" to a few segments that have two levels of streets below them.)

In order to populate this field, it was often necessary to split segments on either side of an overpass, because it was found that if a segment with a Level of "1" were intersected by a segment with a Level of "0," the resulting transition didn't match up exactly right, with the symbology used. However, it was found that the transition between a Level "1" segment and a Level "0" segment, without an intersecting street present, did look okay (the transition is indistinguishable), so this became the desired geometrical configuration.

Figures 46 and 47 on the next page show a typical area that shows the benefit of employing the STRLEVEL field, with appropriate symbology.

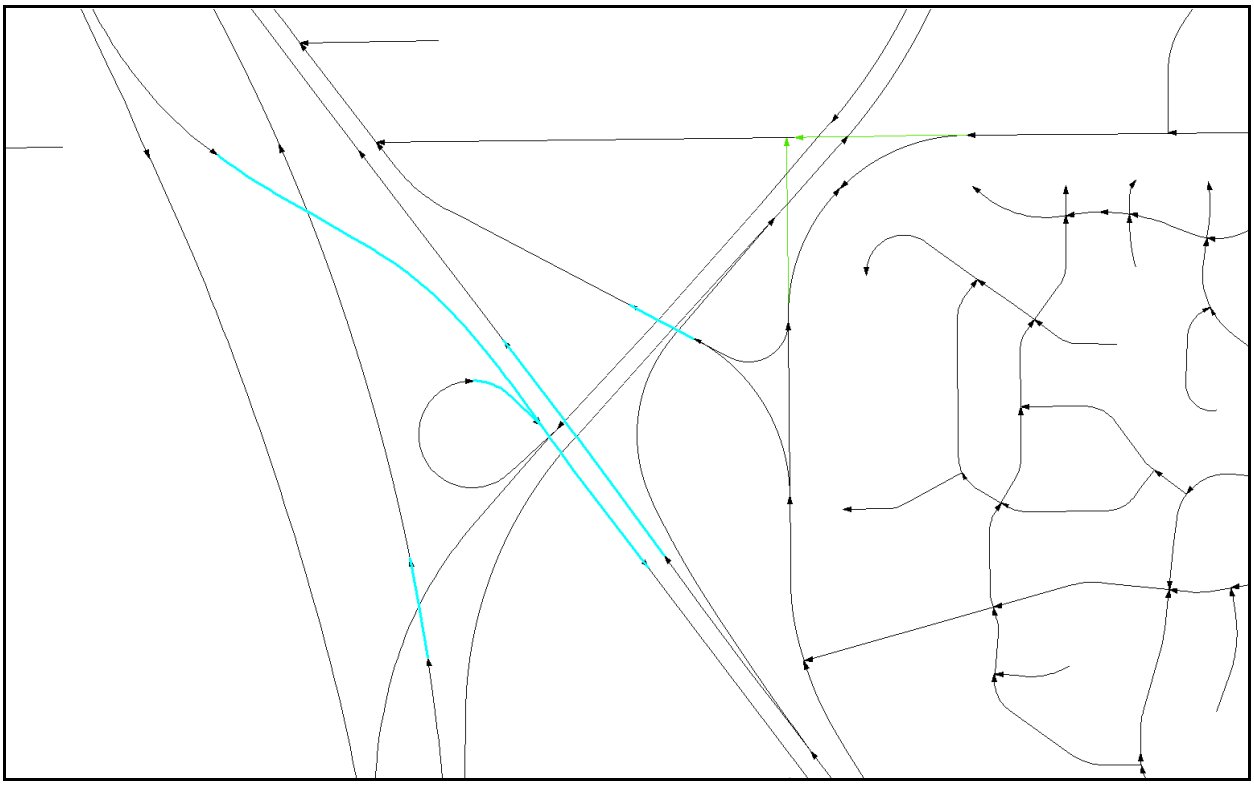


Figure 46 - Example of how the STRLEVEL attribute is used. Segments in blue have been assigned a STRLEVEL value of "1," all others are "0."

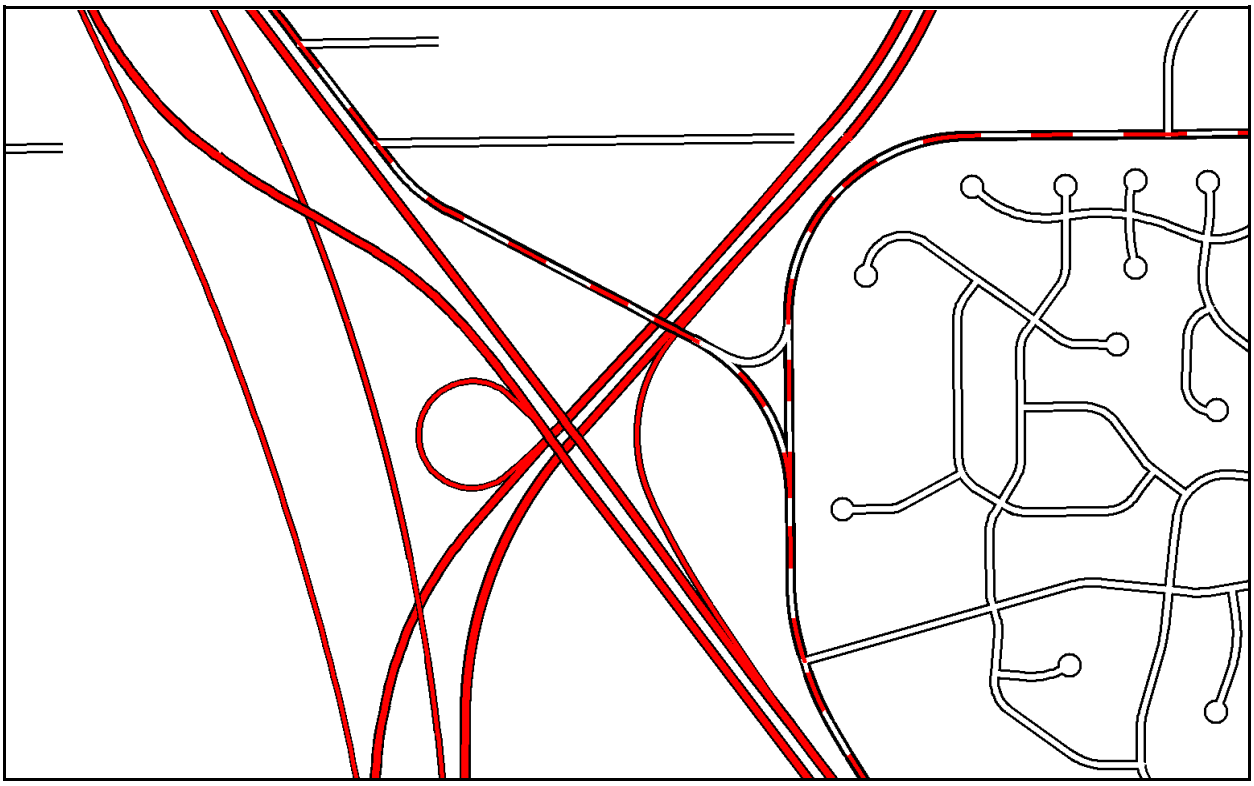


Figure 47 - The same area symbolized using the *Streets With Casings* Layer File, which makes use of the STRLEVEL attribute. Note the realistic-looking overpasses.

The *Streets With Casings.lyr* Layer File

In order to produce a display similar to the one in Figure 47, it is necessary to display various subsets of the centerlines on top of each other, in the proper order, and symbolized appropriately. In our own case, it requires 29 layers! Because this type of symbology is so complicated, I created a Group Layer of all the affected layers, and saved the configuration as a Layer File I call *Streets With Casings*. The purpose of this Layer File is to try and duplicate the look of the 1:24000-scale USGS topo maps (at least as far as the streets are concerned).

My goal for the Layer File was to output street symbology that would actually draw the width of the lines for the streets equal to the approximate real width of the pavement. In other words, if I turn on our *Pavement* layer, which is a line layer representing the edge of pavement for our streets, the distance between the edges of pavement, as shown in this layer, would match closely the distance between the edges of pavement as depicted by the *Streets With Casings* Layer File. Furthermore, this would always have to be true, no matter what scale you zoom to, for displaying or printing. In order to produce this effect, you must assign a reference scale to the Data Frame that contains the Layer File. To create the Layer File, I decided to pick a nice round number for my reference scale, 1:5000, zoom to that scale, and begin creating the various symbols for the various types of streets, with the intent that the line widths would match the actual edge of pavement. That is how I arrived at the various line widths for the different types of streets. (The same is true for the cul-de-sacs. Cul-de-sacs are drawn with the *CuldeSac* layer. See page 79.)

I created a separate Map Document to contain the Layer File (*Streets With Casings.mxd*), so that I wouldn't have to constantly keep turning the reference scale on and off in my Map Document, in order to make the symbols display properly. Thus, whenever I want to print a map using the symbology of the *Streets With Casings* Layer File, I simply open the *Streets With Casings.mxd* Map Document.

One thing I decided when I created the Layer File was that, most of the time when I would be producing a map using this symbology, I would want to included railroads in the map. Thus the Layer File contains our *Railroad* layer. I can always turn that layer off individually in the Layer File, if I want to for a particular purpose.

Another thing I realized was that when using this symbology, it normally wouldn't make sense to include "paper" streets and cul-de-sacs (those not yet built). So, in the definition query for the individual layers, if the street classification includes paper streets (which is all the classifications except Interstates and Ramps), the query would have to be written to exclude the paper streets. The same thing is true for cul-de-sacs.

On the next page is a table outlining the configuration of each of the individual layers in the *Streets With Casings* Layer File. The layers are listed in the table in the same order they are stacked in the Layer File; the layers draw from the bottom up.

Structure of the *Streets With Casings* Layer File

Layer Name	Symbol	Color	Width	Source Layer	Definition Query
Interstates Level 2 Fill	Solid line	Red	5.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Local Streets Level 2 Fill	Solid line	White	4.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Ramps Level 2 Fill	Solid line	Red	3.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Ramps Level 2 Casing	Solid line	Black	5.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Interstates Level 2 Casing	Solid line	Black	8.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Local Streets Level 2 Casing	Solid line	Black	7.0	fcStreets	STRLEVEL = '2' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Railroads	Railroad symbol	Black	5.0	Railroad	N/A
Interstates Level 1 Fill	Solid line	Red	5.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Highways Level 1 Fill	Solid line	Red	5.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'B' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Major Streets Level 1 Fill	Solid line	Alternating Red & White	3.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'C' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Local Streets Level 1 Fill	Solid line	White	4.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Ramps Level 1 Fill	Solid line	Red	3.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Ramps Level 1 Casing	Solid line	Black	5.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Interstates Level 1 Casing	Solid line	Black	8.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Highways Level 1 Casing	Solid line	Black	8.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'B' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Major Streets Level 1 Casing	Solid line	Black	7.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'C' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Local Streets Level 1 Casing	Solid line	Black	7.0	fcStreets	STRLEVEL = '1' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Highways Level 0 Fill	Solid line	Red	5.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'B' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Major Streets Level 0 Fill	Solid line	Alternating Red & White	3.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'C' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')

Structure of the *Streets With Casings* Layer File (Cont.)

Layer Name	Symbol	Color	Width	Source Layer	Definition Query
Local Streets Level 0 Fill	Solid line	White	4.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Cul-De-Sacs Fill	Solid circle	White	15.0	fcCuldeSac	OPER_STATUS = 'BUILT'
Interstates Level 0 Fill	Solid line	Red	5.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Ramps Level 0 Fill	Solid line	Red	3.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Ramps Level 0 Casing	Solid line	Black	5.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'F' AND OPER_STATUS = 'BUILT'
Highways Level 0 Casing	Solid line	Black	8.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'B' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Major Streets Level 0 Casing	Solid line	Black	7.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'C' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Local Streets Level 0 Casing	Solid line	Black	7.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'D' AND (OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT')
Interstates Level 0 Casing	Solid line	Black	8.0	fcStreets	STRLEVEL = '0' AND STRCLASS = 'A' AND OPER_STATUS = 'BUILT'
Cul-De-Sacs Casing	Solid circle	Black	16.0	fcCuldeSac	OPER_STATUS = 'BUILT'

Notes:

For the line fill symbols, I had to use a cartographic line instead of a simple line symbol. This was so I could specify a line cap type of "Square" in the Symbol Property Editor, to make sure the fill pattern would overlap sufficiently where different line levels come together. For the line casing symbols, I used a cartographic line with a "Butt" type line cap.

For the fill symbol for major streets, I used a cartographic line symbol with two layers. The upper layer, which consists of the white dashes, used eight "squares" in the Symbol Property Editor to make up the dash, as well as eight squares to form the gap. I used an "internal" of 4.00. The bottom layer formed the red dashes, using the same number of squares as the upper layer, except the dashes and gaps were reversed. It also used an internal of 4.00.

STRSUBTYPE - When we switched to the geodatabase model for our centerlines, we decided to implement subtypes. We thought we would want to take advantage of the ability to assign different sets of default values and different domains to the different subtypes, and maybe eventually implement connectivity and/or relationship rules, but as of this writing, we have not done this.

The STRSUBTYPE field defines the subtype type. The 12 subtypes are:

Code	Description
1	**Interstate (Built)**
2	**Interstate (Non-Built)**
3	**Highways (Built)**
4	**Highways (Non-Built)**
5	**Major Streets (Built)**
6	**Major Streets (Non-Built)**
7	**Local Streets (Built)**
8	**Local Streets (Non-Built)**
9	**Alleys (Built)**
10	**Alleys (Non-Built)**
11	**Ramps (Built)**
12	**Ramps (Non-Built)**

Since we have not assigned different domains or default values based on our subtypes, I have to remember to update the various other attributes if I change a centerline subtype.

SPD_LIM - Speed limit, in miles per hour. Populated for all streets within Marion County, and interstates outside Marion County. The original source for most of this information was the *Consolidated Code of the City of Indianapolis* about ten years ago. Updates in the form of official Ordinances are sent to me by Carol McAdams of the Traffic Engineering Section of DPW out at Sherman. The default for most streets not specifically cited is 30 MPH, except for streets within the City of Lawrence, which is 25 MPH. Apartments and similar developments carry a default of 15 MPH. Streets outside Marion County, except interstates, carry a value of zero. I have revised a few streets, including the interstates within I-465, based on my own personal observations. Domain follows.

0
10
15
20
25
30
35
40
45
50
55
65

SECONDS - The number of seconds it takes to traverse the specified segment, traveling at the speed limit contained in the SPD_LIM field. (The field is not populated for segments that contain a zero speed limit.) This field was created to be used as an impedance for routing applications, including ESRI's original Network Analyst. Unfortunately, the geodatabase does not allow for automatically-calculated fields, like some databases do. So, when we worked in the coverage world a few years ago, I used to use a macro to calculate the values in this field. Since we switched to the geodatabase model, I have not maintained this field, because I'm not aware of anyone who currently

wants to use it. However, it would be easy to write a script that would update the values in this field. We in Indianapolis have discovered, though, that when we update the values in a field for the entire database, it takes a very long time — too long except to run the process overnight. And this is just for one field!

The formula for calculating the number of seconds in the field is

[SHAPE.LEN] * 15 / 22 / [SPD_LIM]

SPD_LIM_ORD - The number of the City/County Ordinance that established the speed limit contained in the SPD_LIM field, if known. I use a standard way of entering the values in this field. Here is an example: "G.O. 104, 1990," which stands for "General Ordinance #104 of the year 1990." Relatively few of the records in the database contain a value for this field.

ONE_WAY - Indicates the direction of travel on one-way streets. The direction is specified as either from the "from-node" to the "to-node," or vice-versa. The "from-node" and "to-node" are defined by the direction of digitization. If a street is two-way, the value in this field will be blank (or [Null]). This field was created to be used by ESRI's original Network Analyst for routing applications.

The original source for most of this information was the *Consolidated Code of the City of Indianapolis* about ten years ago. Updates in the form of official Ordinances are sent to me by Carol McAdams of the Traffic Engineering Section of DPW out at Sherman. Domain follows.

FT - The direction of travel is from the "from-node" to the "to-node."
TF - The direction of travel is from the "to-node" to the "from-node."

ONE_WAY_DIR - This field also indicates the direction of one-way streets. It was created to give the user an intuitive indication of the direction of travel; a "user-friendly" version of the information in the previous field. The values in this field are the four main directions of the compass. For streets running diagonally, the direction assigned is the one that most closely aligns with the direction of the segment. For curved segments or segments with turns, the direction assigned is the one most closely aligned with the direction of the segment at its end ("to-node"). For example, imagine a cloverleaf interchange, and imagine you're traveling the loop (ramp) that takes you from eastbound to northbound. This loop will be coded "NORTH," because that is the direction you would be traveling when you leave the loop.

The value for two-way streets will be blank (or [Null]). Domain follows.

NORTH
SOUTH
EAST
WEST

ONE_WAY_ORD - The number of the City/County Ordinance that established the street as a one-way street, if known. I use a standard way of entering the values in this field. An example is "G.O. 104, 1990," which stands for "General Ordinance #104 of the year 1990." Obviously, for interstates and highways, although they may be one-way streets, this field doesn't apply, because they would have always been one-way. Relatively few of the records in the database contain a value for this field.

NAME_CHANGE - This field holds supporting documentation for the renaming of the street from the name listed in the OLD_NAME field (if known). The preferred documentation is the official Resolution number, but if that isn't known, a date (general or specific) is also acceptable. In the past, DMD used to convene what was called the Address Advisory Committee, of which I was a member, that was responsible for distributing notices (Resolutions) of street renamings. The committee quit

meeting several years ago. Since then, I have occasionally received copies of Resolutions from Sherry Powell out at Sherman. Fortunately, street renamings are rare, so it will probably be easy to keep up with new changes in the future.

Some time ago I received from Brian Schneider copies of old lists of streets that have been renamed in the last 150 years or so. Maybe if I live to be 150 years old I will be able to transpose all that information onto the centerlines! In the meantime, I sometimes get information on old street names from copies of Vacation Petitions that I receive when streets get vacated. In these cases, I know a street name has changed, but that's all I know, so I will enter the old name in the OLD_NAME field, and enter "UNKNOWN" in the NAME_CHANGE field. That way, I will know the information wasn't simply overlooked when these attributes were populated.

Relatively few of the records in the database contain a value for this field, even for streets that I know have been renamed (which is practically all the streets within a two-mile radius of downtown).

VACATED - This field is intended to hold information to support that a street has been vacated. The preferred information is a Vacation Petition number, but a date (general or specific) is also acceptable. There are several methods I use to determine if a street has been vacated, which have varying degrees of reliability:

1. I receive a copy of the Vacation Petition. Currently these are delivered to me by Cecilia Capers of DMD. I have been receiving these for about eight years now. Sometimes these Petitions will also show other streets in the vicinity that were previously vacated. All this information I transfer to the centerlines. This is the most reliable source of vacations.

2. When I look at the centerlines superimposed against the *Parcels* layer, sometimes it is apparent from the parcels that a street or alley used to exist, but has been vacated.

3. A guy named Ed Hazelrigg used to work for Indianapolis DOT (which is now part of DPW), and he kept records on street vacations. I have a set of old paper DMD basemaps that Ed colored in to show vacation locations. All of these I have transposed to the centerlines. I also have a Professional File database of street vacations of Ed's, which I have not attempted to transfer over, because of the amount of time it would take. (Professional File is an old flatfile PC database program that was popular in 80's.)

If a street is vacated but I have no supporting documentation, I enter "UNKNOWN" in this field, so that if I revisit the centerline, I know I didn't simply just overlook that information.

PRE_DIR - Street name pre-directional, as defined by the *Address Guidelines and Standards*, published by DMD. In Indianapolis, we have developed addressing standards to fit our own particular street names. This includes how we parse our street names. We have identified four potential parts of every street name:

- Pre-directional
- "Kernal"
- Suffix
- Post-directional

Not every street name contains all four parts. In fact, the only required part is the kernal.

I will not attempt to go into a complete explanation here of our addressing standards, as that would be lengthy. That information is all contained in the *Address Guidelines and Standards* document. However, for street names that require a pre-directional, here is the domain:

N - North
S - South
E - East
W - West

STREET_NAME - The "kernal" of a street name (that portion of a street name after any pre- and post-directional and suffix is stripped off). This field must be populated for every centerline.

For interstate names, the convention used is a capital "I" followed by a hyphen, and then the number of the interstate (e.g., "I-74"). For United States highways, the convention is to use the letters "US," without periods or a space between, followed by a space, and then the highway number (e.g., "US 52"). The convention is the same for State highways, except we use the letters "SR" (for "State Road"). For all these types of streets, the entire description goes in the STREET_NAME field.

For streets within Marion County that are part of a U.S. or State highway, if the streets goes by a local name, the local name will be the one used in the STREET_NAME field, and the highway number designation will be listed in the ALIAS_FULL_STNAME field (see page 69). Example: STREET_NAME = "MERIDIAN ST," ALIAS_FULL_STNAME = "SR 135."

For most streets, the street name is not in question, but there are a significant number of streets for which the street name is questionable. For example, some streets go by more than one name. One name might be an old name for the street still in common usage, or it might be an alias, which is another currently accepted name for the street (see the ALIAS_FULL_STNAME field, below). For a few streets, it seems "impossible" to determine the official name. This is due to the fact that information on old streets is usually time-consuming to research, and official records may be sketchy or non-existent. Records may even conflict. I have also found that some streets were never "officially" named, but may have been given a name through common usage.

Generally, in the absence of better information, I consider the *Street Location Guide* published by DMD to be the most accurate source of street names when centerline street names are in question. I will also occasionally consult other commercial maps (both new and old), DMD Basemaps, and old plats to help resolve questionable street names. Another good source is the scanned historical maps we have available on our server. In all cases, if I find more than one plausible name for a street, I will put my best guess in the FULL_STNAME field (with the "kernal" in the STREET_NAME field), and all the other possibilities in the ALIAS_FULL_STNAM field.

Besides actual street names, there are two more possible values for the STREET_NAME field:

UNKNOWN - I use this value if the street name is unknown to me at the time the centerline is created, but I am reasonably sure the street does have a name. In some cases, the name of the street could be ascertained with a bit of checking (consulting maps, plats, etc.), and in other cases, it might be impossible, or take a lot of work to find out the street name. If I don't have the time to research a street name (which is usually the case), I will assign UNKNOWN to the street name. Later, if I discover the name, I will fill it in.

UNNAMED - I use this value for streets I think don't have a name. These are usually minor private streets, possibly connecting an apartment or commercial complex to a thoroughfare, or connecting two complexes together. If I later find out the street does have a name, I will fill it in.

Ramps

The names of the interstate ramps are those assigned by INDOT. Originally I obtained these from a document given me by an INDOT employee. The document is a collection of diagrams of all the interstate interchanges in Marion County. Each interchange has the ramps labeled with their respective designation. Here is an example of a typical ramp name:

I-465 033 RAMP D

This ramp name follows a standard convention that INDOT uses to identify ramps. The first part of the name is the name of the interstate. The second part is the interchange number. Interchanges are assigned a number which is the same as the closest mile marker of the interstate. Since interchange numbers may have as many as three digits, if an interchange number is less than 100, I assign a leading zero to the number, so that the numbers (and consequently the street names) will sort correctly.

The third part of the name is the word "RAMP," followed a letter of the alphabet, which identifies the individual ramp.

Certain letters of the alphabet are reserved for particular ramps configurations. (This was never "officially" explained to me, but is something that became apparent to me after examining several interchange diagrams.) If a particular interchange does not contain any ramps of a particular configuration, then the letter assigned to that ramp type will not be used for any of the ramps for that interchange.

For example, letters "A" through "D" are reserved for the four possible outermost ramps of a typical cloverleaf interchange, one in each quadrant. The letters are assigned in a counter-clockwise direction, normally starting in the southeast quadrant. (However, for interchanges involving two different interstates, it appears the letters are assigned beginning in the southwest quadrant, for some reason.)

The next four letters, "E" through "H," are reserved for the four possible loops of a typical cloverleaf. Again, the letters are assigned in a counter-clockwise direction, beginning with the same quadrant that the first of the outermost ramps is assigned.

The rest of the letters of the alphabet are assigned to any remaining ramps. I have been unable to determine if a standard exists for the assignment of these remaining letters. Some interchanges display the letters "P" and "R" through "T" at the four corners of the interchange bridge, but I am not sure exactly what these refer to. Figure 48 on the next page is an example of a typical interchange, showing how the ramps are assigned letters. (All Figures depicting ramps are from the collection of interchange diagrams I obtained from INDOT.)

If a particular ramp configuration is absent at a particular interchange, the letter for that particular ramp is simply skipped when assigning letters to the ramps. For example, Figure 49 on the next page shows an interchange that contains only two of the possible four outermost ramps.

Collectors and distributors are also assigned letters. Figure 50 two pages over shows such an interchange.

Some interchanges are interchanges between two interstates. Since each interstate has its own mile marker numbering system, these types of interchanges will have two different numbers, depending on which interstate numbering system you are referring to. Therefore, the ramps at these interchanges will have two different names. Figure 51 two pages over is an example of such an interchange. Each ramp in this interchange has two names. For example, the southeast loop can be designed as either I-465 016 RAMP F, or as I-74 073 RAMP F.

INDOT has now posted the interchange diagrams on the Internet at www.in.gov/dot/pubs/manuals/interchanges/. This site replaces the old collection of paper diagrams, so from now on, this is where I will go to get designators for revised interchanges.

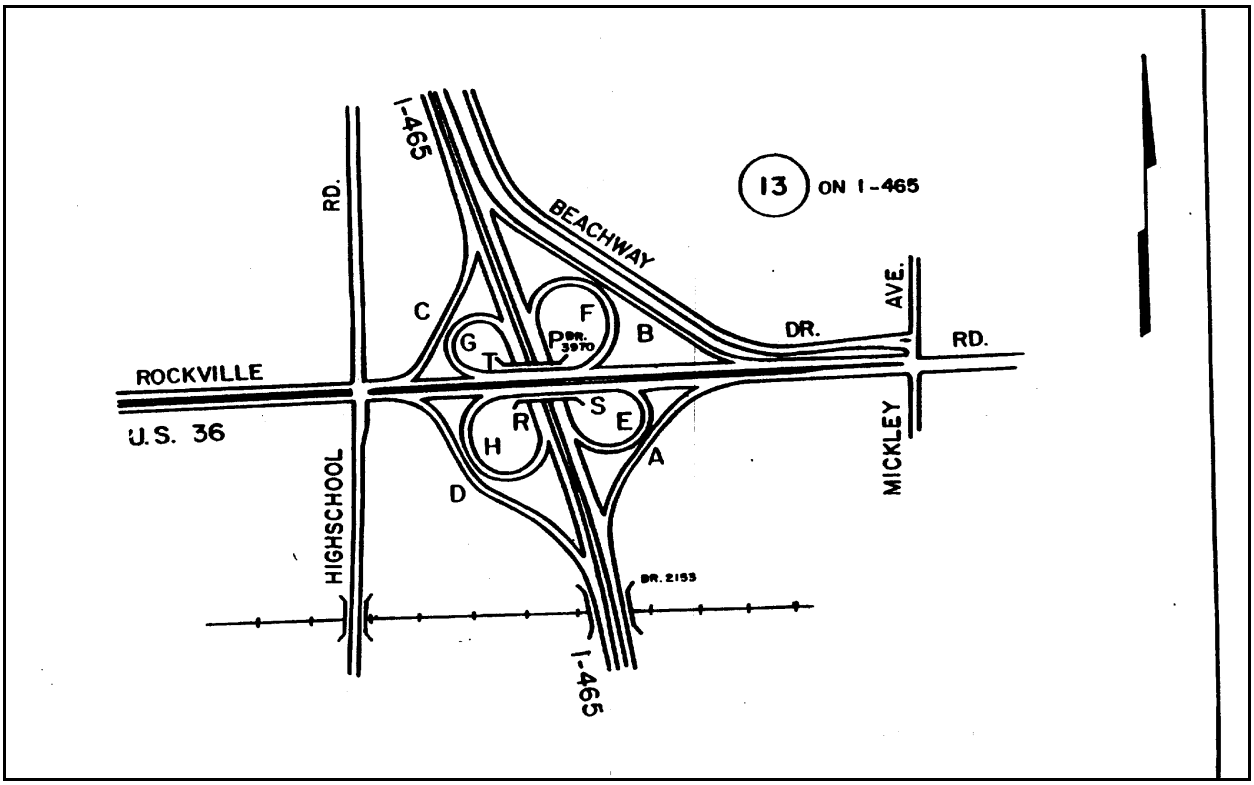


Figure 48 - Typical cloverleaf interchange.

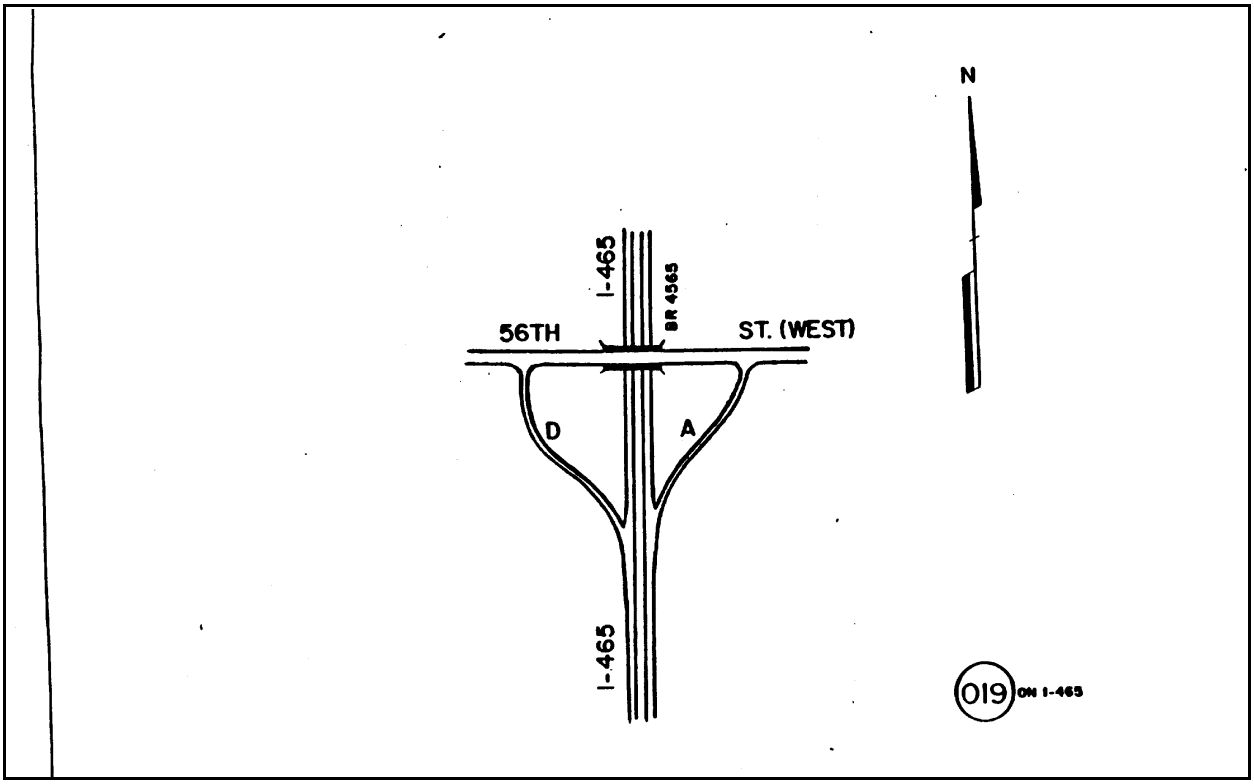


Figure 49 - Interchange showing how designator letters are skipped when certain ramp configurations are not present.

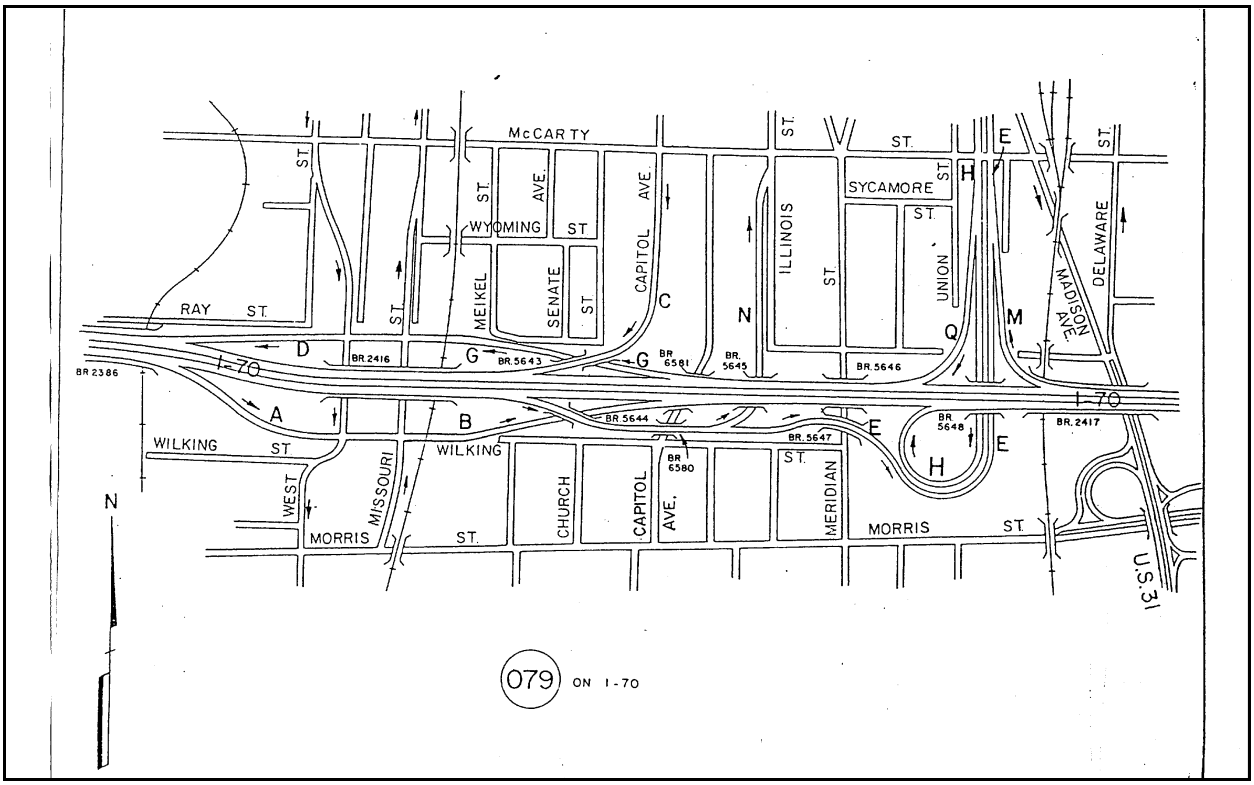


Figure 50 - Interchange diagram showing assignment of designators to collectors/distributors and other miscellaneous ramps.

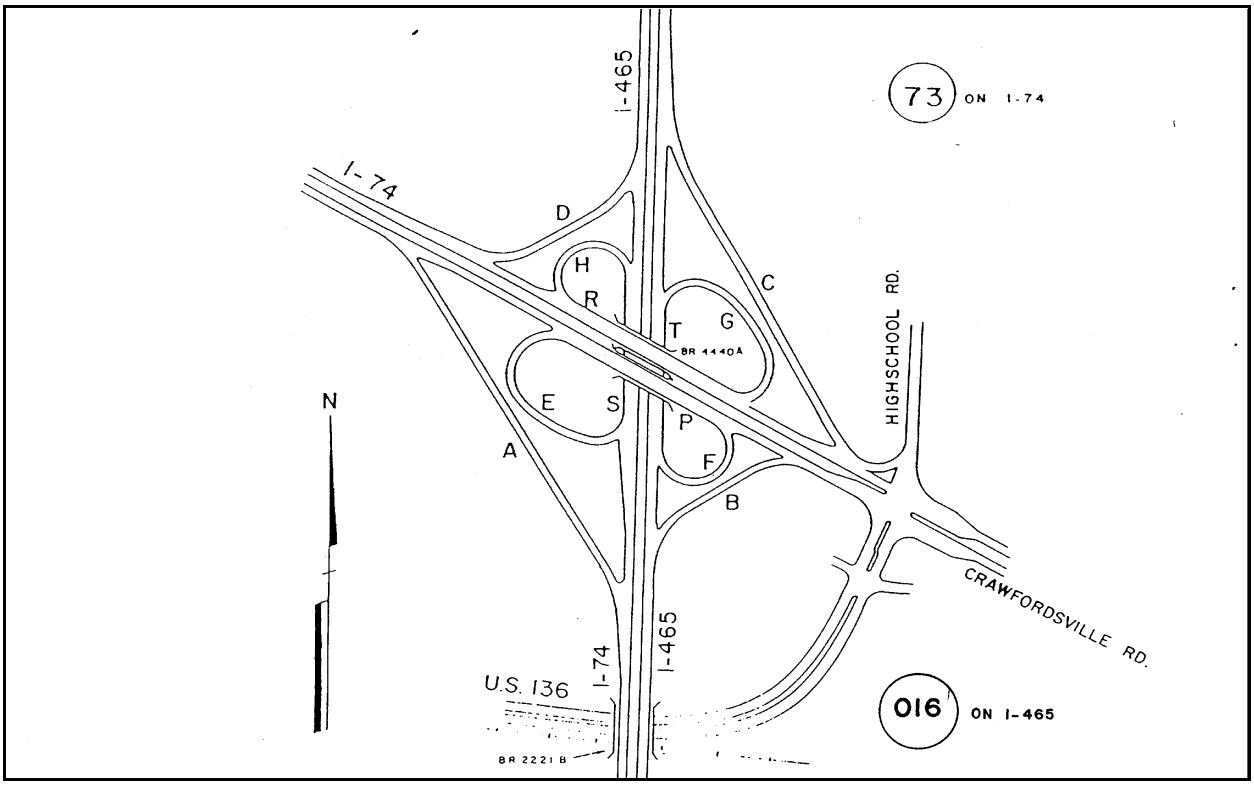


Figure 51 - Typical interchange of two interstates.

Alleys

Alleys are identified from our *Parcels* layer. They are easy to spot, because of their narrow right-of-way. For my purposes, I define any pathway with a right-of-way of less than 24 feet to be an alley.

Alleys with a generally accepted, "official" name, are identified by that name on the centerlines (e.g., Muskingum Street). Alleys without a common name are named according to a convention adopted a number of years ago by a committee I served on, called the City/County GIS Committee. This is basically equivalent to the methodology used to name county roads (e.g., 200 E).

Alleys are assigned a coordinate based on the Marion County address grid system. The address coordinate of an alley is derived from the coordinates of the nearest streets on either side of the alley, and not from the parcel addresses at the particular alley location. For example, consider an alley located halfway between two streets whose coordinates are 2700 west and 2800 west. The parcel address on the street running perpendicular at that location might be 2714 west (because parcel addresses are not usually assigned proportionally between intersecting streets), but the alley will be assigned a coordinate (and name) of 2750 west. If the alley were one-quarter of a block west of 2700 west, it would be assigned a coordinate of 2725 west, regardless of what the parcels in that area were addressed. See Figure 52, next page, for a real life example.

Generally I try to stick to multiples of 25 when assigning alley coordinate names, but this is not always possible, due to the complex arrangement of some alleys. In all cases, however, the coordinates will always be multiples of five.

For the purposes of the naming convention, all alleys must be considered either "north-south" or "east-west" alleys, so for alleys running at an angle, I try to decide which direction predominates when assigning the name. When deciding what coordinate name to assign to an alley, if the alley only runs for one block, I will pick an approximate address near the middle of the alley segment. If the alley runs for more than one block, I will pick an average, representative address for every two, three or four blocks of the alley's length, and assign the same name to all those segments. That way, the name of the alley won't be changing in each block. See Figure 53, next page.

The names of alleys (other than those with common names) always start with the word "ALLEY," followed by the approximate address coordinate, and then a post-directional, indicating which side of the address grid the alley is on.

Here are a couple of examples of alley names:

E ALLEY 2425 S
N ALLEY 620 W

When the alley centerlines were first created, I did not know which alleys would exist on both sides of the address grid meridians. Thus, to be safe, I assigned a pre-directional to all alley names. Someday, I need to go through all the alleys and remove the pre-directional from the ones that should not have one (according to the *Address Guidelines and Standards*).

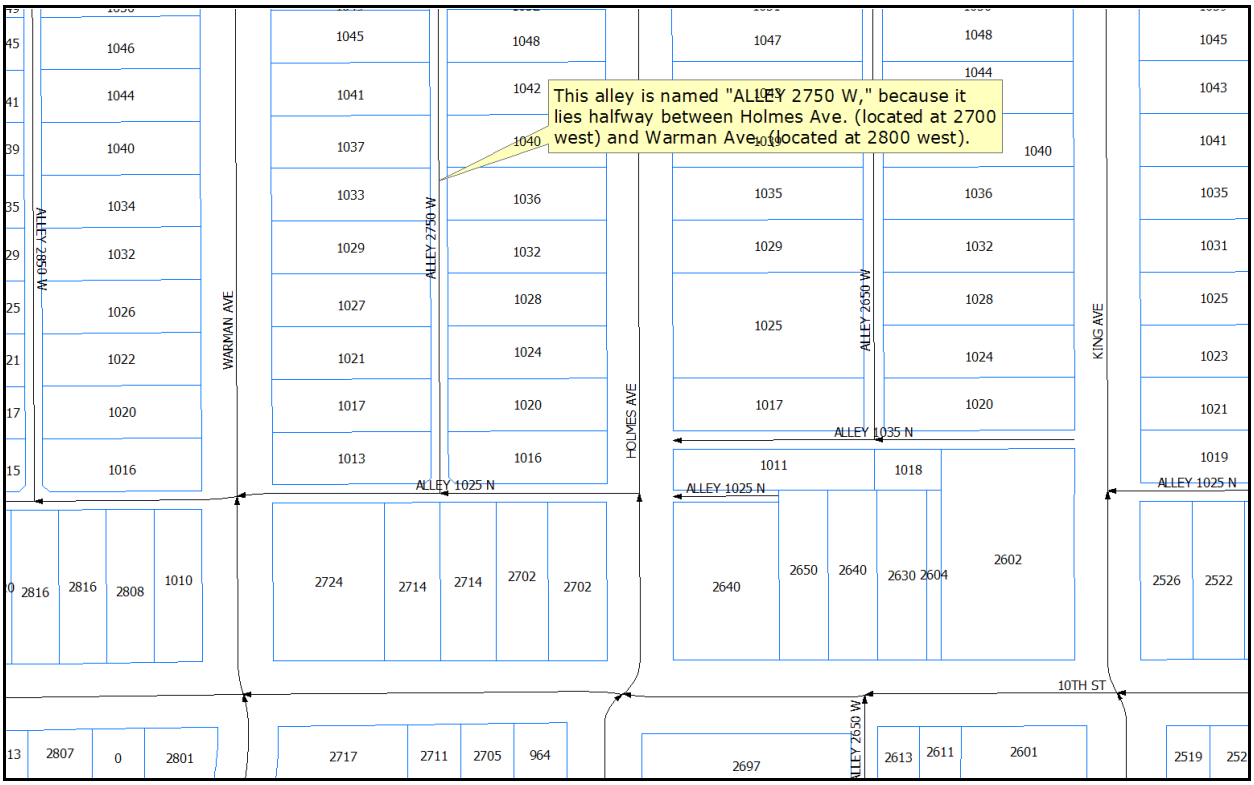


Figure 52 - Notice how alleys are named with respect to the address coordinates of their adjacent streets, not the addresses of their adjacent parcels. (The pre-directional is not included in the alley labels.)

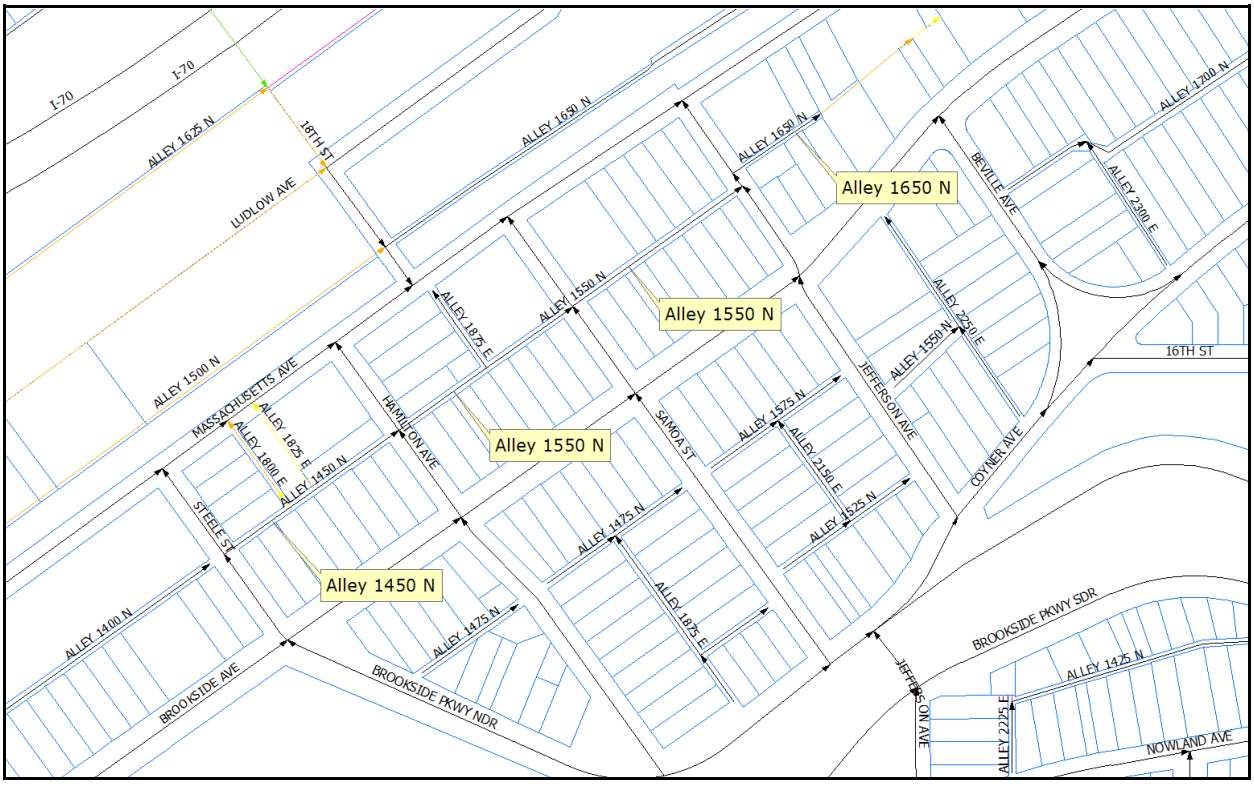


Figure 53 - Notice how ALLEY 1550 N keeps its same name for the two block stretch between Hamilton Ave. and Jefferson Avenue, instead of being assigned two different names for each block (pre-directionals not shown).

STREET_TYPE - Street suffix. Domain follows. Per the *Address Guidelines and Standards*, certain combinations of words are abbreviated as one suffix, in order to shorten the street name in our databases. Note that not every street name contains a suffix.

AVE - Avenue
BLVD - Boulevard
CIR - Circle
CT - Court
DR - Drive
EDR - East Drive
ELN - East Lane
LN - Lane
NDR - North Drive
PKWY - Parkway
PL - Place
RD - Road
SDR - South Drive
ST - Street
WAY - Way
WDR - West Drive
WLN - West Lane

SUF_DIR - Suffix directional (post directional). Domain follows. Note that not every street name contains a post-directional.

N - North
S - South
E - East
W - West

FULL_STNAME - The value of the full street name field is equal to the concatenation of the values in the PRE_DIR, STREET_NAM, STREET_TYP, AND SUF_DIR fields, in that order, with only one space between each of the parts. If a street name doesn't contain all the parts, leading and trailing spaces are dropped. This field must be populated for every centerline.

STR_LABEL - Street label. This field will either contain the same value as the FULL_STNAME field (with one exception — see below), or it will be blank. This field is intended to be used as the source field in most cases when employing auto-labeling for making maps. The field is left blank in certain instances so that certain centerlines won't be labeled. These are streets where it generally doesn't make sense to label them, and they include:

- Alleys.
- Ramps.
- Streets for which the value in the FULL_STNAME field is "UNKNOWN" or "UNNAMED."
- Very short segments, usually those for which the value of the NO_ADDR field is "INT," or sometimes "NOA."
- For divided highways and interstates, only one centerline of the divided pair will have this field populated. This is so you don't produce redundant labels for these streets. In these situations, the same side of the street will contain the field with the STR_LABEL value throughout its entire length, as a convention. For example, for an interstate running north and south, only the west centerlines for the interstate might have this field populated.

If it is desirable to label all the streets on a map, the FULL_STNAME field can always be used in place of the STR_LABEL field.

The one exception when the value in this field will differ from the FULL_STNAME field is for streets whose name starts with a number less than 10, such as "09TH ST." Although a leading zero is inserted in the name of these types of streets in the STREET_NAME and FULL_STNAME fields, in the STR_LABEL field, the leading zero is dropped. This is because we want to use the "human-readable" form of the street name when labeling maps, instead of the "computer-readable" form. The computer-readable form is desirable for sorting the street names.

ALIAS_FULL_STNAME - This field is used to hold one or more street name aliases (other names the street is known by, besides the one in the FULL_STNAME field). A street segment can have an unlimited number of aliases, and if there is more than one, they will be separated by semicolons in this field. The alias names will include the entire street name, just like the FULL_STNAME field. Sometimes, I will place a note in the REMARKS field to help explain the derivation of an alias. Here is a list of types of aliases that might be included in this field. (Note that former official names of a street go in the OLD_NAME field, not this field.)

- Alternate names, where a street goes by more than one name, and both names appear to be official (example: Henry St./Wayne Ave. just east of Lynhurst Drive).
- Alternate names, when the official records of a street name are in conflict, or it is difficult to determine if the street has been renamed, and one of the names of the street is a former name (example: 58th St./58th St. S. Dr. between Keystone Ave. and Rural Street).
- Alternate names, where a street is commonly thought of as having a certain name because of its alignment compared to adjacent street segments, even though technically the street has a different official name (example: Allisonville Rd. as an alias for Temple Ave. between Binford Blvd. and 45th Street).
- State highway designations (example: US 52).
- County road names (example: 900 W).
- References to old alignments of streets or highways (example: Old US 431).

ADDRESSING_GRID - This field was created to support our *Street Guide* report. When we list all our streets and what address they begin and end at, we recognized a few areas where streets are addressed using a different address grid than Marion County's. These include the excluded cities of Beech Grove and Southport, and the platted subdivision of Eagledale. For streets addressed using these unique grids, this field contains the name of that area. All other records contain a null value. When the *Street Guide* report utility is run, it will output the values in this field for the appropriate streets, so that the reader knows the addresses are not from the Marion County address grid.

This field only applies to streets within Marion County. Domain follows.

BEECH GROVE
SOUTHPORT
EAGLEDALE
[NULL]

WEIGHT_LIM - Official weight limit, in pounds, if a street has one. Only applies to streets in Marion County. Weight limits are assigned by Ordinance. I populated this field from information contained on the website www.municode.com, which contains the Revised Code for the Consolidated City of Indianapolis/Marion County. If a street doesn't have a weight limit, the value of this field will be null. I have requested that Carol McAdams of DPW out at Sherman send me copies of all future Ordinances that revise weight limits, so that I might keep this information current. Domain follows.

6000
10000
11000
16000
20000
[NULL]

WEIGHT_LIM_ORD - Ordinance number of the Ordinance that established a weight limit on a segment. Only applies to streets in Marion County. I did not populate this field when the WEIGHT_LIM field was populated, because it was impossible to determine from the www.municode.com website which Ordinance number applied to the weight limits that are already established. Besides, this would have involved a lot of work anyway. However, for new weight limit Ordinances that I receive, I will populate this field.

BLOCK_ID - This field is a key field used to join the centerlines with the Pavement Management System (PMS) database administered by Sherry Powell out at Sherman. The "Block_ID" is the identifier her database uses to identify a particular centerline. Being able to join the two databases enables us to make maps that show pavement conditions and ratings.

Years ago DPW had an intern take a copy of the centerlines and use it to populate the CENTERLINE_TAG field in the PMS, which had been added to the database in order to do the join. Then Sherry sent me the table with the populate attribute, and I in turn used it to populate the BLOCK_ID field on the centerlines a year or two ago.

Unfortunately, there is currently no procedure in place to maintain this attribute. We do not want to give Sherry edit rights to the centerlines, just to maintain one field. So it is left up to Sherry and I to try and maintain this attribute manually. And further hindering the process is the fact that Sherry doesn't have much time to devote to sending me updates. Sherry's system is the one that assigns new BLOCK_ID'S, so I have to get that information from her. And when I split or merge centerlines, Sherry has to receive that information from me. So for the time being, the two databases are getting farther and farther out of synch. Hopefully in the future we can come up with a method to maintain this link.

DATE_ACCEPTED - About a year ago this date field replaced the old text field called simply "ACCEPTED." The purpose of that field was to hold supporting documentation pertaining to the acceptance of a street into DPW's street maintenance program. For most streets for which the field was populated, which was mainly the streets that have been built in the last ten years, this value was simply the date from the Completion and Compliance Affidavit I receive from George Krack. For a few other streets, maybe only the year (or even the decade) was known, so that would also go in that field.

The problem with this field was that it was a text field, and consequently I could not do date arithmetic with the data. I have a need to be able to find all the streets where the current date is greater than or equal to three years since the acceptance date, and where the MAINT_JURIS value is DEVELOPER. (See the discussion of the DEVELOPER value on page 46.) So I had this new date field created to satisfy that requirement. I moved all the date values over from the ACCEPTED field, and then the ACCEPTED field was deleted. For those values that were not a complete date, or that were miscellaneous text, I put the values in the REMARKS field.

Relatively few of the records in the database contain a value for this field. That is because this field has only been maintained for the last few years, and it would be very time-consuming to go back and populate the information for older streets (assuming we even have that information).

OTHER LAYERS

Associated Layers

The *Intersection* Layer

The *Intersection* layer is a point layer showing the locations of logical street intersections. (The concept of logical intersections was introduced on page 13.) This layer only contains those intersections within (or on the border of) Marion County.

Just like centerlines, intersections are never deleted — their attributes are used to show which ones are active. The attributes on the intersections as they exist today are listed in the table below, in the order they appear for editing. A description of the attributes follows.

Intersection Attributes

Field Name	ORACLE Type	Length	Allow Null Values	Default Value	Domain Name	Precision	Scale
INTERSECTION_TAG	Double	-	Yes	0	-	16	-
COMPTYPE	Long Integer	-	Yes	-	-	5	-
COMPKEY	Long Integer	-	Yes	-	-	9	-
UNITID	Text	15	Yes	-	-	-	-
TYPE1	Text	20	Yes	-	-	-	-
TYPE2	Text	20	Yes	-	-	-	-
TYPE3	Text	20	Yes	-	-	-	-
DATE_CREATED	Date	-	Yes	-	-	-	-
DATE_MOVED	Date	-	Yes	-	-	-	-
DATE_CHANGED	Date	-	Yes	-	-	-	-

INTERSECTION_TAG - Just like for centerlines, this is a unique number used to identify the intersections. Tag numbers are assigned sequentially, and the lowest number is 50021. (We started with 50000, because it's a nice round number, but the first 20 intersections were created in error, while we were testing the procedure.) New numbers are assigned automatically by the *CMFI* ("Custom MAD Feature Inspector," see page 133) just like for centerlines. Numbers have currently surpassed the 93000 figure. When a tag number is retired (the intersection is deleted, because it was created in error), it is never reused.

COMPTYPE, **COMPKEY** - These fields work exactly like the ones for the centerlines. They are not currently being maintained.

UNITID - Just like for the fields above, this field will also be used to store a value that Hansen's IMS program uses to identify intersections. This field is not currently being maintained.

TYPE1 - This field is used to distinguish between street and alley intersections. The participating streets and alleys may be either built or non-built. Domain follows:

STREET/STREET - The intersection is an intersection of two or more streets. There may also be alleys participating in the intersection, but as long as there are at least two streets present, it is classified as a STREET/STREET intersection.

STREET/ALLEY - The intersection is an intersection of a street with one or more alleys.

ALLEY/ALLEY - The intersection is an intersection of an alley with one or more other alleys.

TYPE2 - This field is used to distinguish between built and non-built intersections. The entities participating in the intersection may be either streets or alleys.

BUILT/BUILT - The intersection is an intersection of two or more built (improved) centerlines. Of all the centerlines participating in the intersection, if at least two of them from different streets or alleys are built, then the entire intersection is classified as "BUILT/BUILT," regardless of how many of the other centerlines of the intersection are not built.

BUILT/NON-BUILT - Only one of the centerlines participating in the intersection is built. It doesn't matter if the built centerline is of a street or an alley; neither does it matter what the non-built centerlines are.

NON-BUILT/NON-BUILT - None of the centerlines participating in the intersection are built.

TYPE3 - This field is used to distinguish between grade and separated intersections. An intersection may not be both GRADE and SEPARATED; if it is, it would be classified as two separate intersections.

GRADE - The intersection is an intersection where all the centerlines are at the same elevation, which means traffic can traverse the intersection (assuming the intersection is built).

SEPARATED - The intersection marks the location where one street passes over another street (or alley). Traffic cannot traverse the intersection (pass from one street to the other).

DATE_CREATED, **DATE_MOVED**, **DATE_CHANGED** - These fields work exactly like the ones for centerlines.

Symbology

I have developed standard symbols that I use to display the intersections in the *Intersection* layer. These symbols are based on the values of the TYPE1, TYPE2, and TYPE3 attributes. The symbology is stored in a Layer File called *Intersections.lyr*. These symbols consist of a central symbol surrounded by a halo. The following table describes the symbols used for each combination of attribute values.

		Symbol Type	Halo	Symbol Color
TYPE1 Value	STREET/STREET	Circle, Size 12		
	STREET/ALLEY	Square, Size 9		
	ALLEY/ALLEY	Diamond, Size 10		
TYPE2 Value	BUILT/BUILT		Orange, Size 3	
	BUILT/NON-BUILT		Blue, Size 3	
	NON-BUILT/NON-BUILT		Magenta, Size 3	
TYPE3 Value	GRADE			Yellow
	SEPARATED			Black

Figures 54 and 55 on the next page are examples of how the intersections look when displayed.

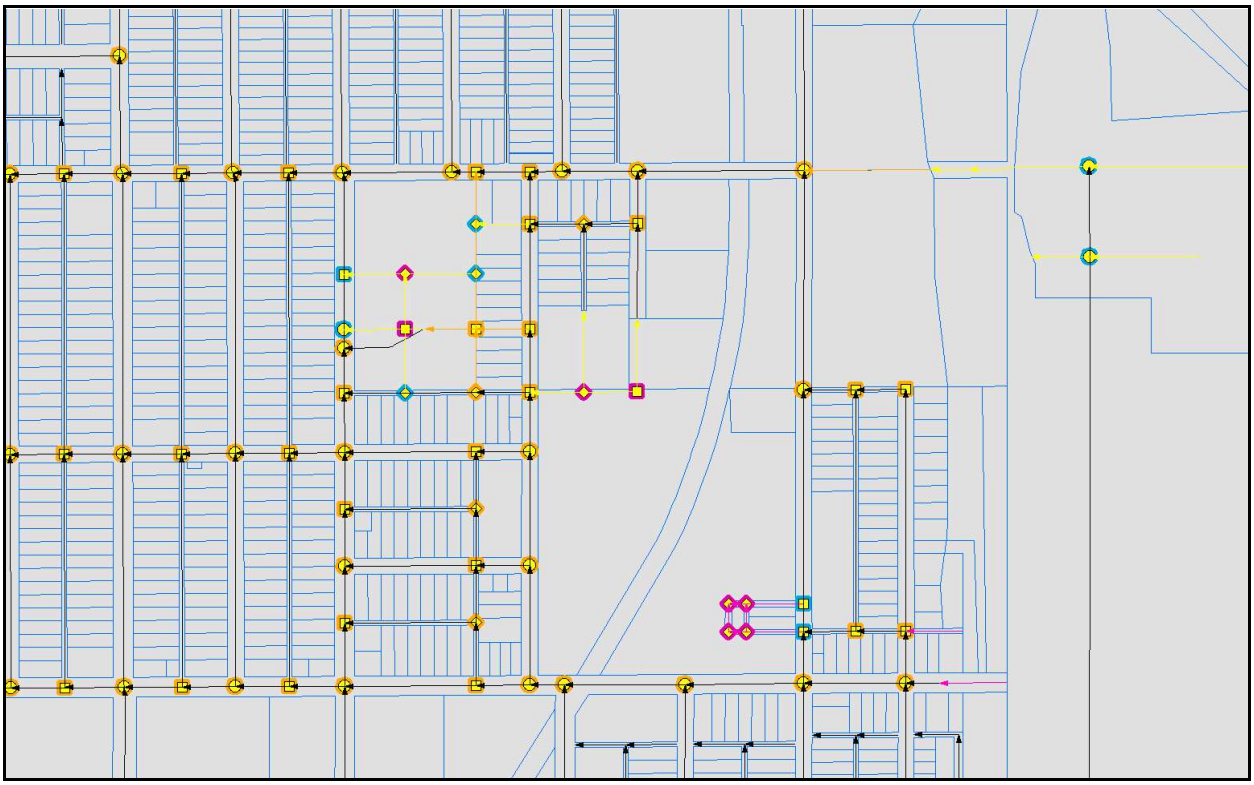


Figure 54 - Intersection symbolization.

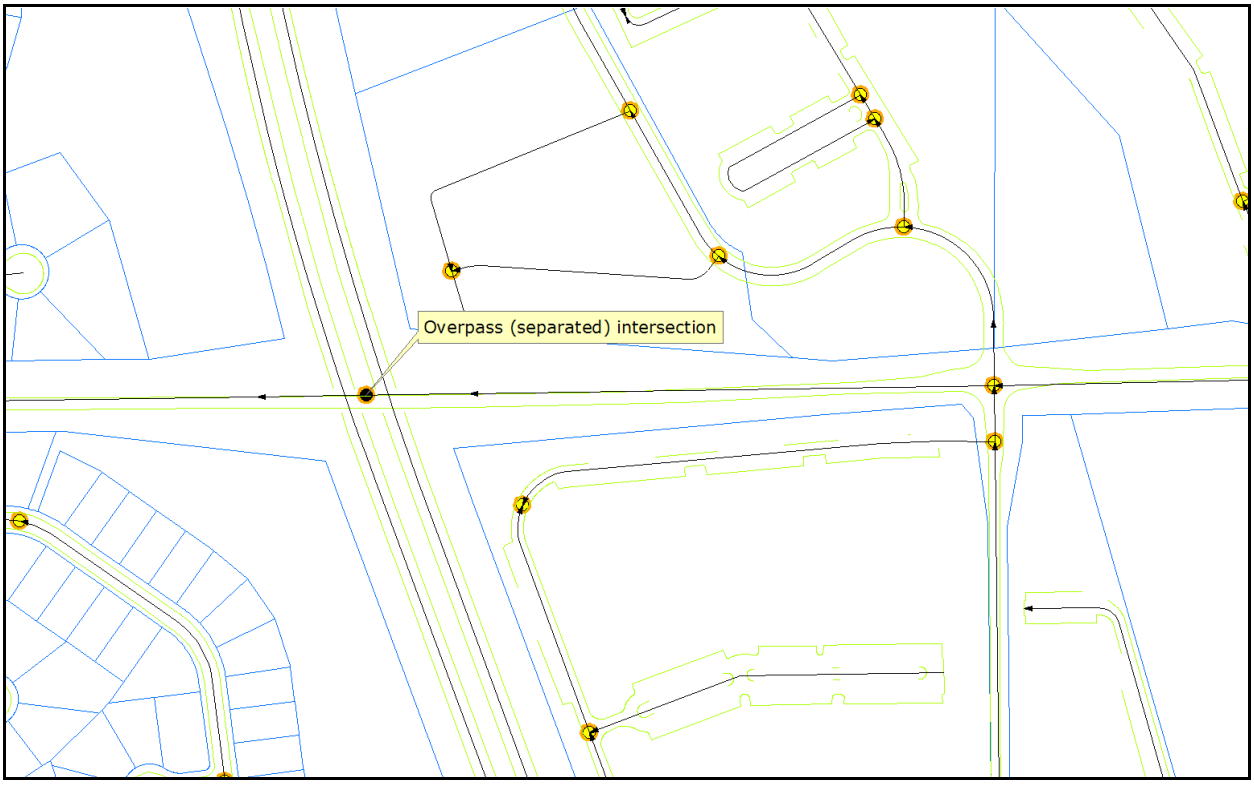


Figure 55 - Example of an overpass intersection.

Logical Intersections

Intersections in the *Intersection* layer are logical intersections. This means that not every intersection of two or more centerlines will have a point in the *Intersections* layer. Consider:

- Sometimes a street will curve, and the addresses on the street change from east-west to north-south at the curve. Although the centerlines will be split where the direction of addressing changes, that point is not a logical intersection, because both centerlines will still have the same street name. See Figure 56, next page.

- There are streets that have little loops that branch off from the side of the street. The intersections of the main street with the loops do not qualify as a logical intersection, unless the street name changes between the loop intersections. A few years ago I checked with Ron Brand of our Traffic Engineering Section of DPW, and he told me that DPW doesn't put street signs at these little "intersections." Therefore, I decided not to treat these as logical intersections, because there would not be any traffic control data potentially linked to them. See Figure 57, next page. Figure 56 shows an example where the street name does change, so a logical intersection is created there.

- Sometimes a street will change names in mid-block. The point where the street changes names is a logical intersection, because there are two different streets coming together there. See Figure 58, two pages over.

- There are places, mostly in private developments, where two streets will cross by way of a little roundabout. If the individual intersections of the roundabout are sufficiently close together, I consider all the intersections making up the roundabout one logical intersection, and I will place the logical intersection point in the center of the roundabout. See Figure 59, two pages over.

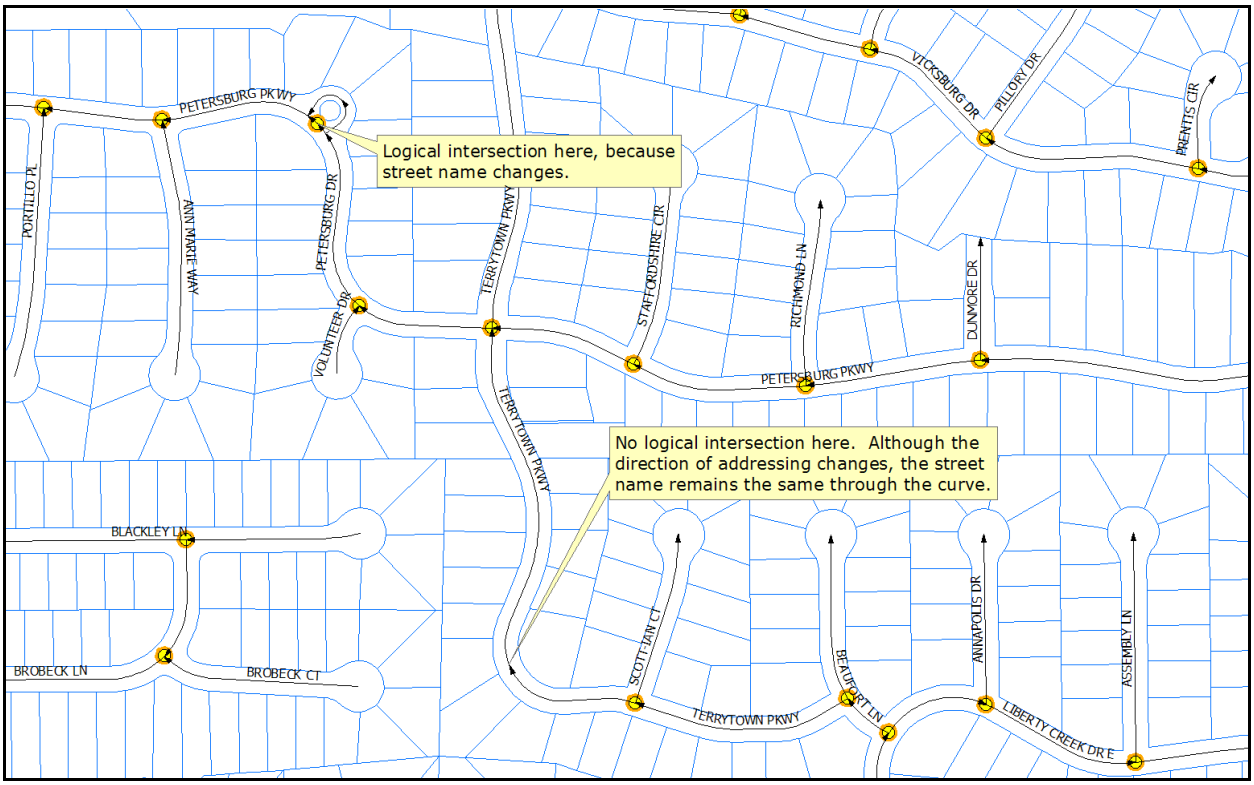


Figure 56 - At points between intersections of centerlines, the street name must change in order for that point to be classified as a logical intersection.

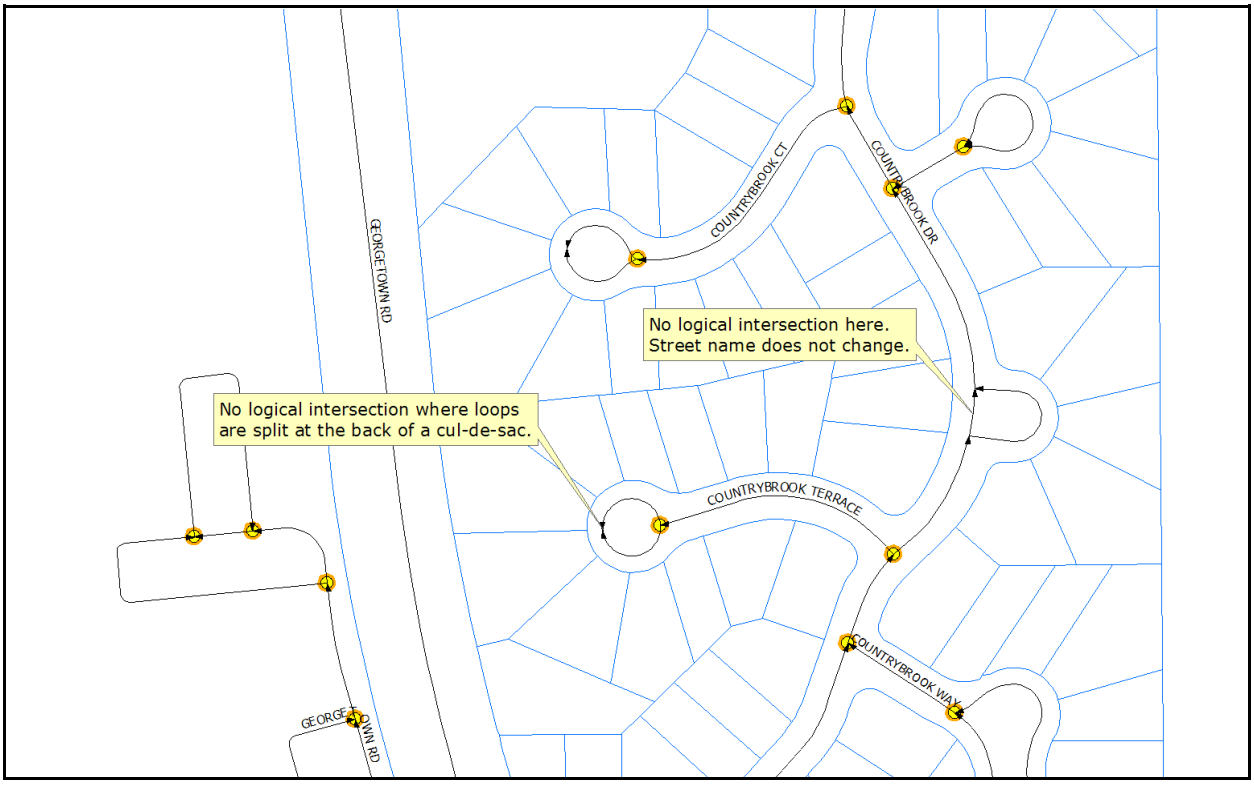


Figure 57 - A couple of situations where a logical intersection is not created, even though the centerlines are broken.

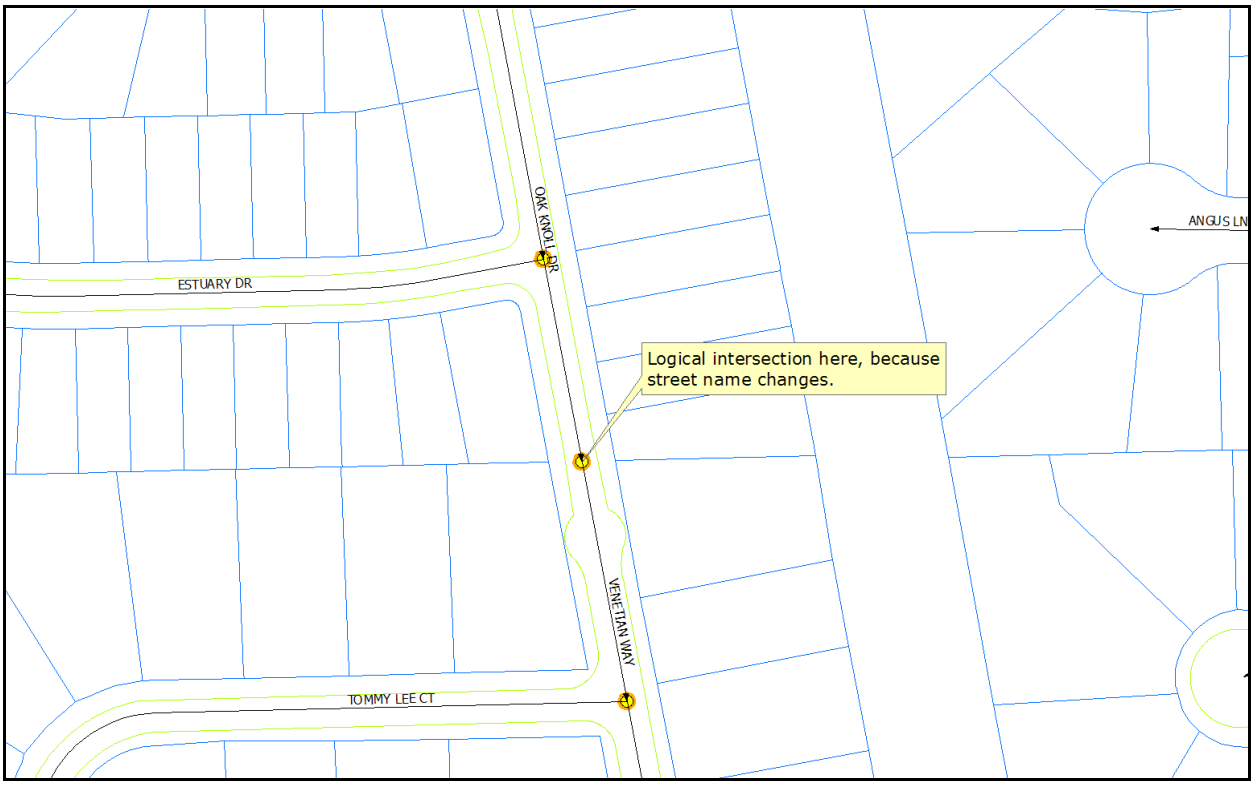


Figure 58 - Example of a logical intersection at mid-block.

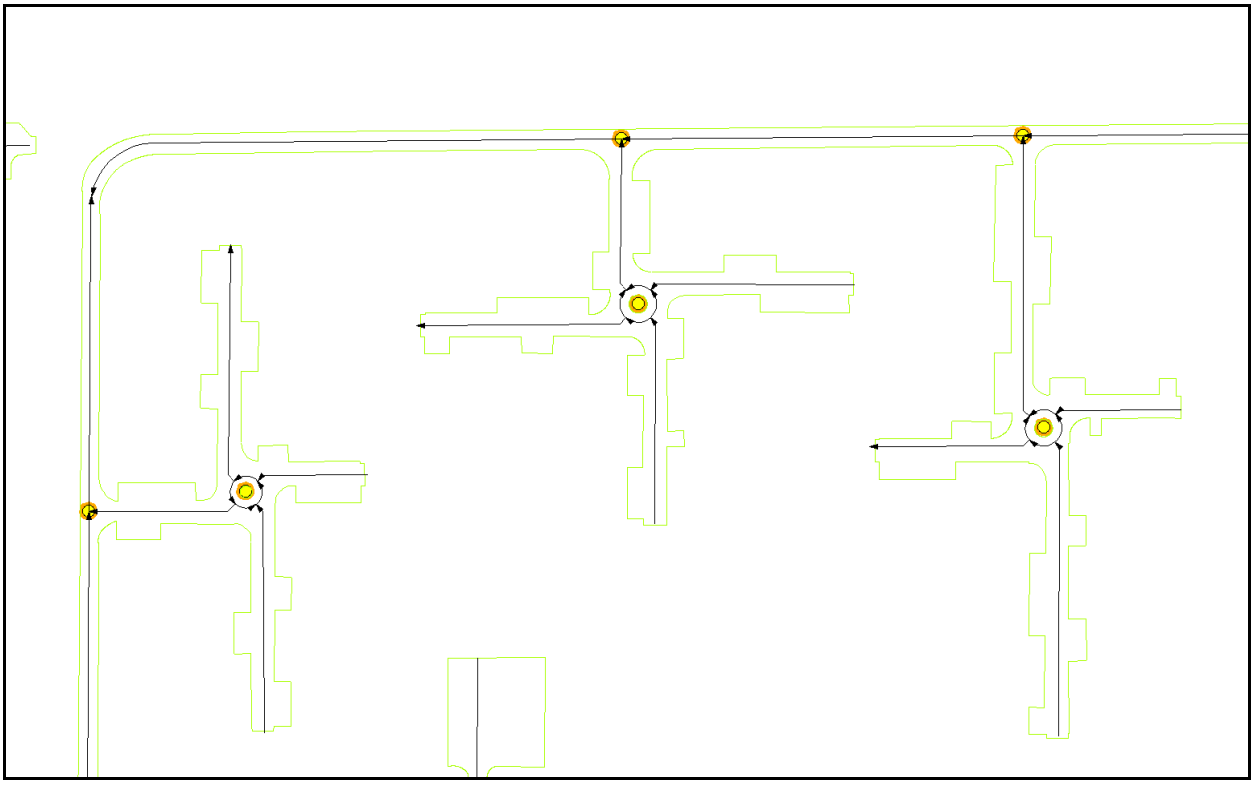


Figure 59 - Sometimes roundabouts are considered one logical intersection.

Associating Intersections With Their Respective Centerlines

The reason that the (logical) intersections don't contain the street names of their intersecting streets is that we decided not to put those attributes on them. Instead, we decided to relate the points to the centerlines by way of a lookup table in the Master Address Database. This results in a more normalized database. The relate is done manually by way of a custom tool that was supplied along with the CMFI, the "Intersection Editor." (See page 175.)

Associating the intersection points with their respective centerlines is not as easy as it may seem. This is because sometimes all the centerlines associated with an intersection do not touch the intersection point. Figure 59 on the previous page is a good example. For those types of intersections, none of the associated centerlines actually touch the intersection point. Another example is shown in Figure 60, below.

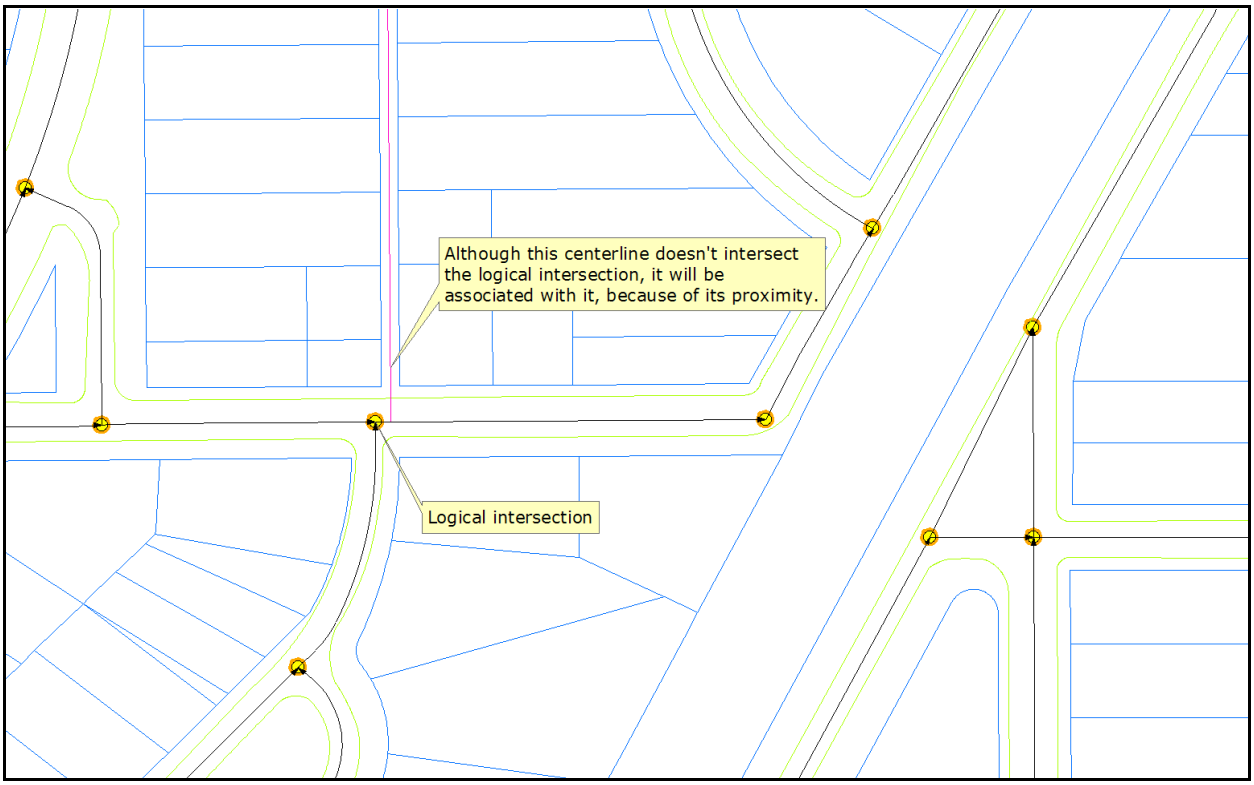


Figure 60 - The platted centerline is associated with the intersection point just above it.

In the example above, because the platted centerline intersects the north-south street so close to the logical intersection point, the platted street will also be assigned to the logical intersection point. Because of its proximity to the intersection of the build streets, it doesn't make sense to create a separate logical intersection for the point where the platted centerline intersects. This illustrates how the concept of logical intersections reduces the number of intersections in the database.

The *CuldeSac* and *CuldeSacOOC* Layers

The *CulDeSac* layer is a point layer, created for cartographic purposes only, showing the location of cul-de-sacs within Marion County. The *CuldeSacOOC* ("cul-de-sacs out-of-county") layer contains the cul-de-sacs for the *StreetsOOC* layer. These layers participate in the *Streets with casings* Layer File (see page 56) to help produce the USGS topo map look.

For the purposes of this document, a cul-de-sac is defined as a piece of pavement at the end of a street, which is wider than the typical cross-section, and whose purpose is to allow vehicles to turn around without having to back up. It is also defined as that portion of right-of-way of a street whose width is wider than the typical cross-section, and whose purpose is to allow for the construction of such a piece of pavement as described above, regardless of whether or not the pavement is actually constructed. All cul-de-sacs will have a point placed in the *CuldeSac* layer at their center.

If a street with right-of-way terminates in a cul-de-sac whose center point is offset from the centerline of the rest of the street, the centerline of the cul-de-sac shall be represented as a circular arc drawn from the center of the cul-de-sac to a point tangent to the centerline of the rest of the street. The tangent point will be the point on the centerline which is perpendicular to the point on the right-of-way where the right-of-way begins to widen for the cul-de-sac. The cul-de-sac point is snapped to the end of the centerline. See Figure 61, next page. Figure 62 shows the result when drawn with casings.

For an example of a cul-de-sac where the right-of-way was not expressly platted to allow for the construction of a cul-de-sac, see Figure 63, two pages over.

Occasionally cul-de-sacs are not located exactly at the end of the right-of-way of a street centerline, but several feed from the end. In this case, the cul-de-sac point is placed at the end of the pavement (in the center of the cul-de-sac), instead of at the end of the right-of-way. See Figure 64, two pages over.

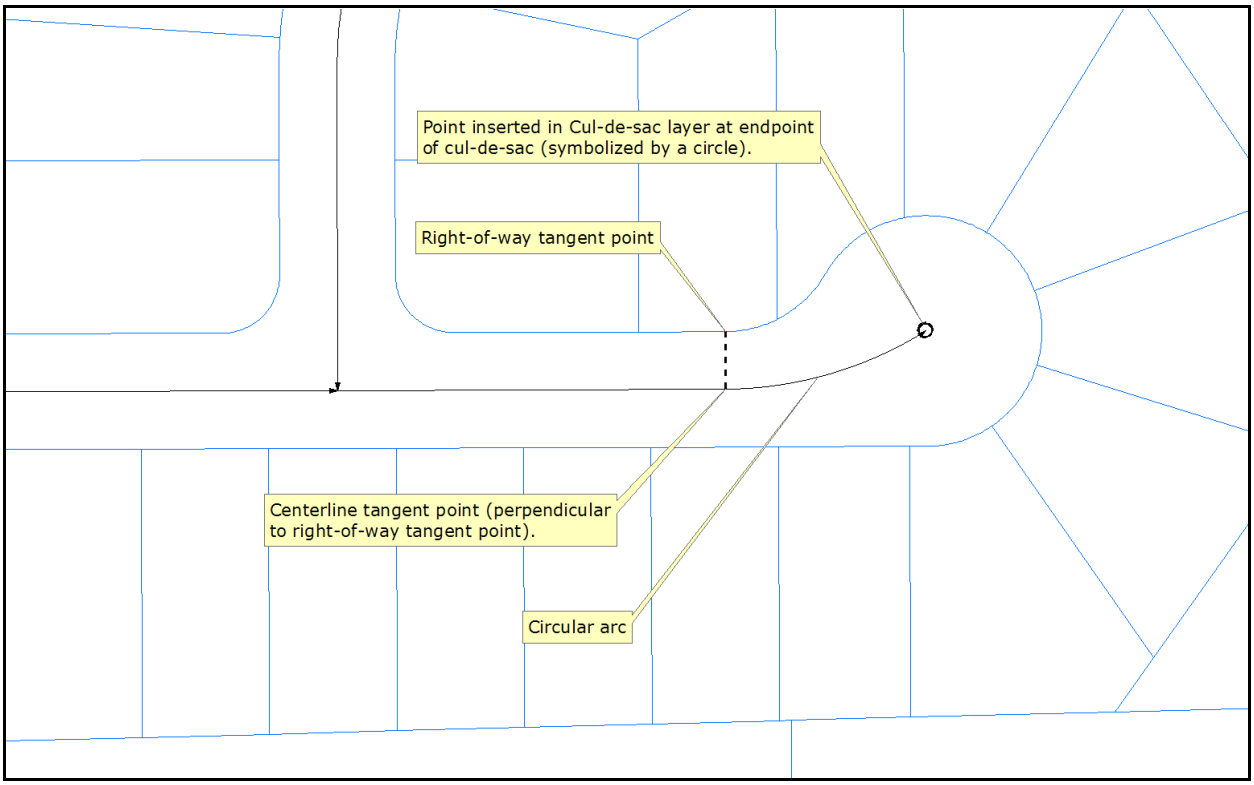


Figure 61 - Typical cul-de-sac whose center point is offset from the centerline of the main part of the street.

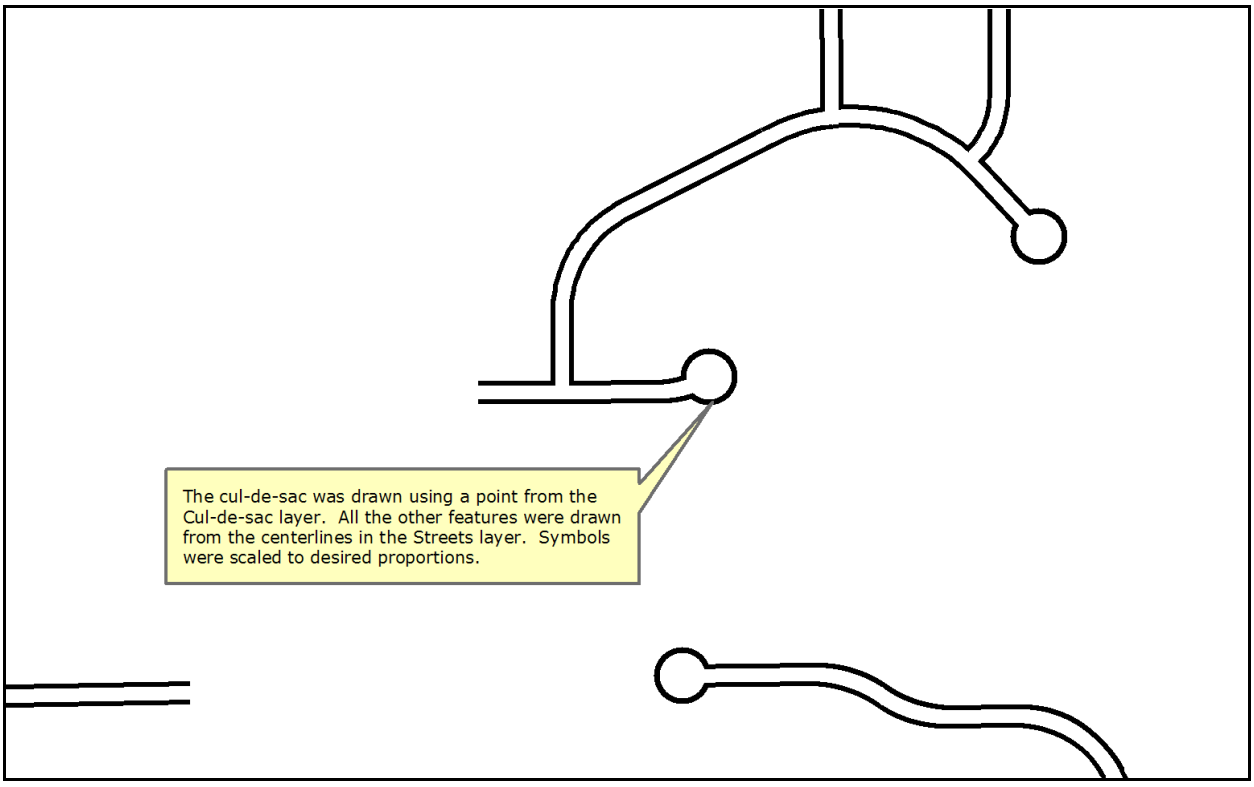


Figure 62 - The same area as the above Figure (zoomed out some), when drawn with casings.

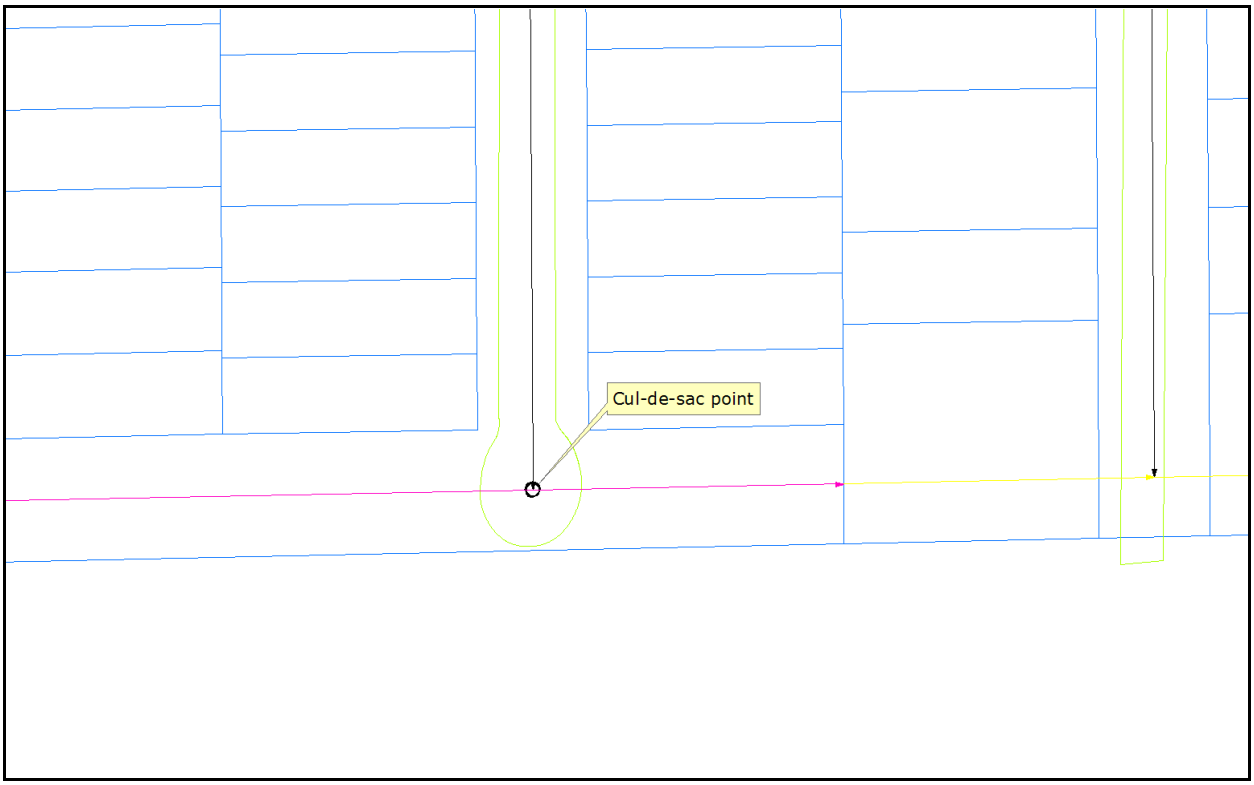


Figure 63 - Example of a cul-de-sac, where the cul-de-sac is not delineated by the right-of-way.

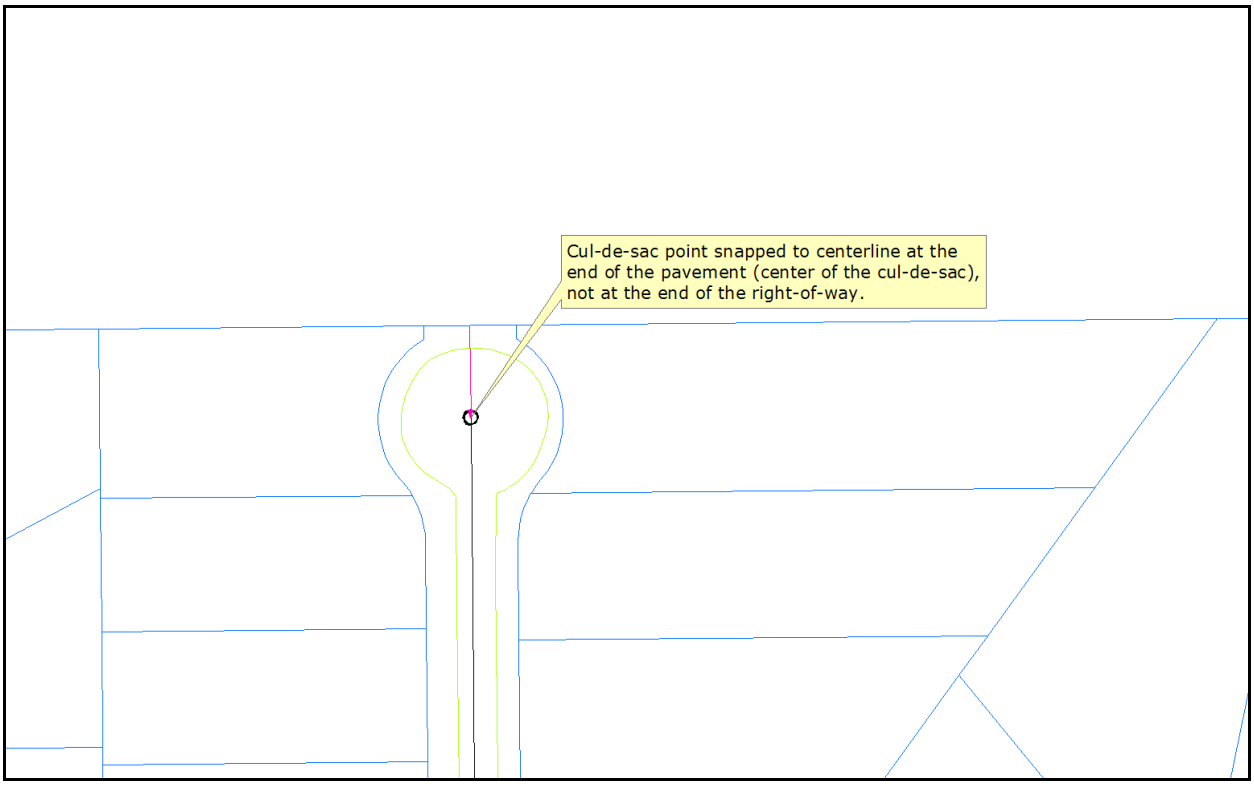


Figure 64 - Typical cul-de-sac not located exactly at the end of the right-of-way of a street.

There are places where the pavement of a street has a widened area, but not located at a dead end. I call these areas "bubbles." Sometimes the purpose of these areas is to allow vehicles to turn around, just like a cul-de-sac. Points are placed in the *CuldeSac* layer at these locations. It doesn't matter whether or not there is right-of-way delineating the bubble area. Sometimes these bubbles are centered on the street centerline, and sometimes they are offset. For an example of the first type, see Figure 65 below.

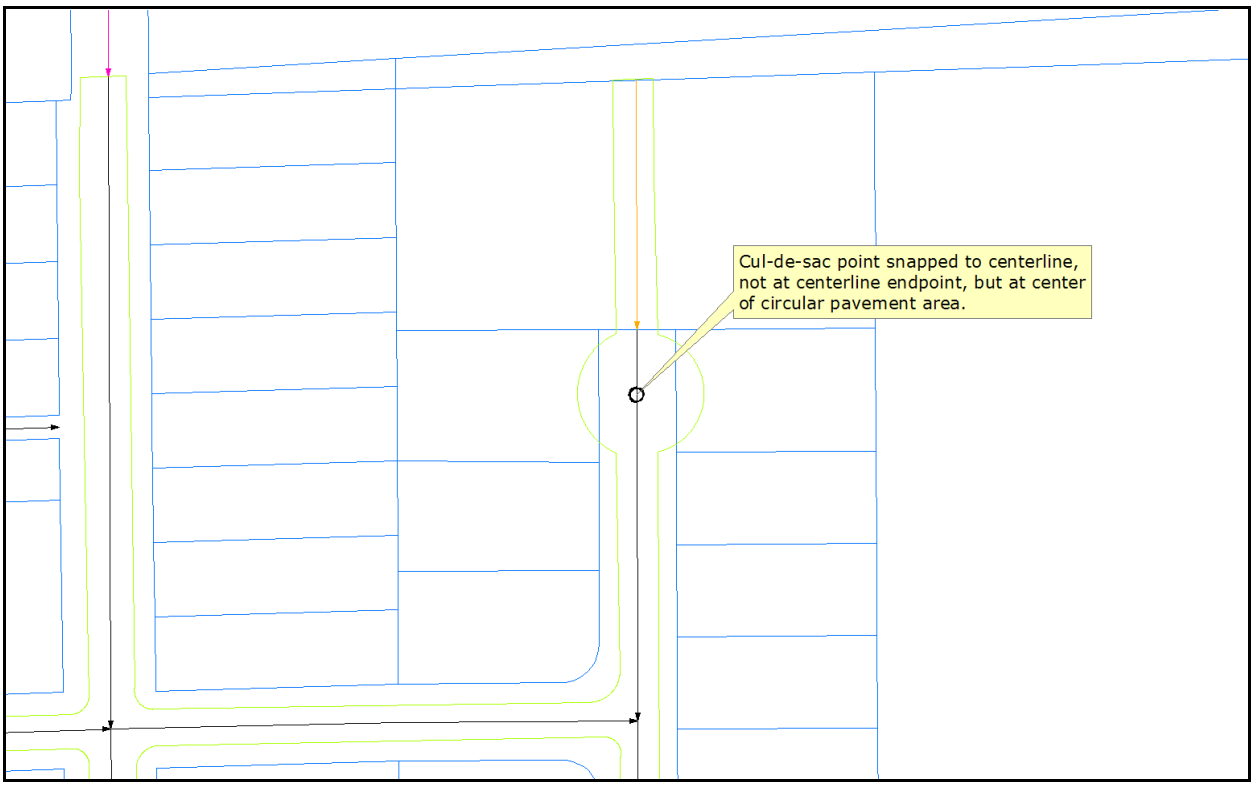


Figure 65 - Example of a pavement "bubble" centered on a street centerline.

In the case where the center of the bubble is offset from the centerline, I place a cul-de-sac point which is offset 29 feet (in map units) from the centerline, and centered within the pavement bubble. Experimentation determined that 29 feet represents the optimum offset for displaying the bubble using my casings layer. To find the desired point location, I construct a temporary centerline 29 feet long, perpendicular to the centerline, and centered with the pavement bubble. After the cul-de-sac point is snapped to the end of this temporary centerline, the temporary centerline is deleted. See Figures 66 and 67 on the next page.

Another common type of pavement bubble is found in subdivisions, where a street turns. This is another case of the cul-de-sac point needing to be offset from the centerline, in order to achieve the desired effect. I experimented with the degree of offset in these situations, and discovered that optimum results were obtained when this cul-de-sac point is offset 24 feet (in map units) from the centerline of the curve. Here I construct a temporary centerline 24 feet long, perpendicular to the curve, and centered within the bubble. After snapping the cul-de-sac point to the end of this line, the temporary line is deleted. See Figures 69 and 68 two pages over to see what this looks like.

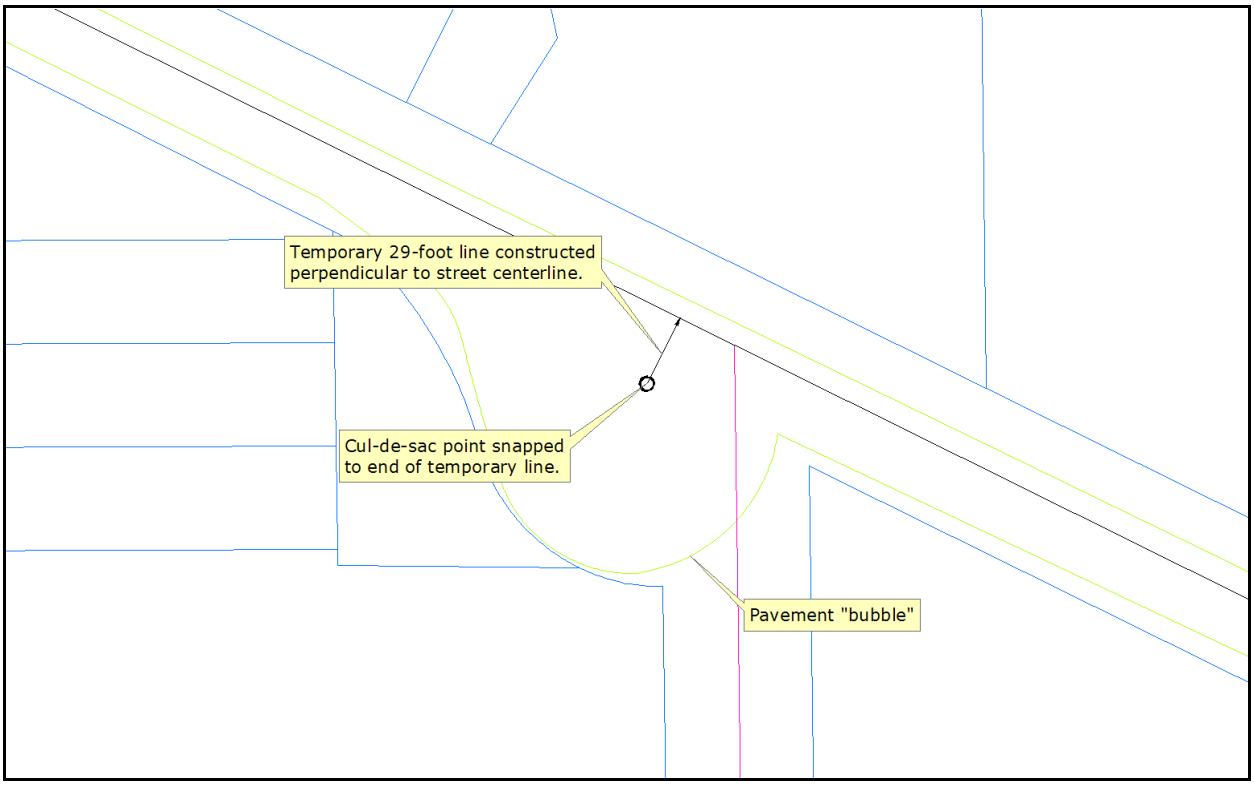


Figure 66 - An example of a "bubble" located in the middle of the street, and whose center point is offset from the centerline.

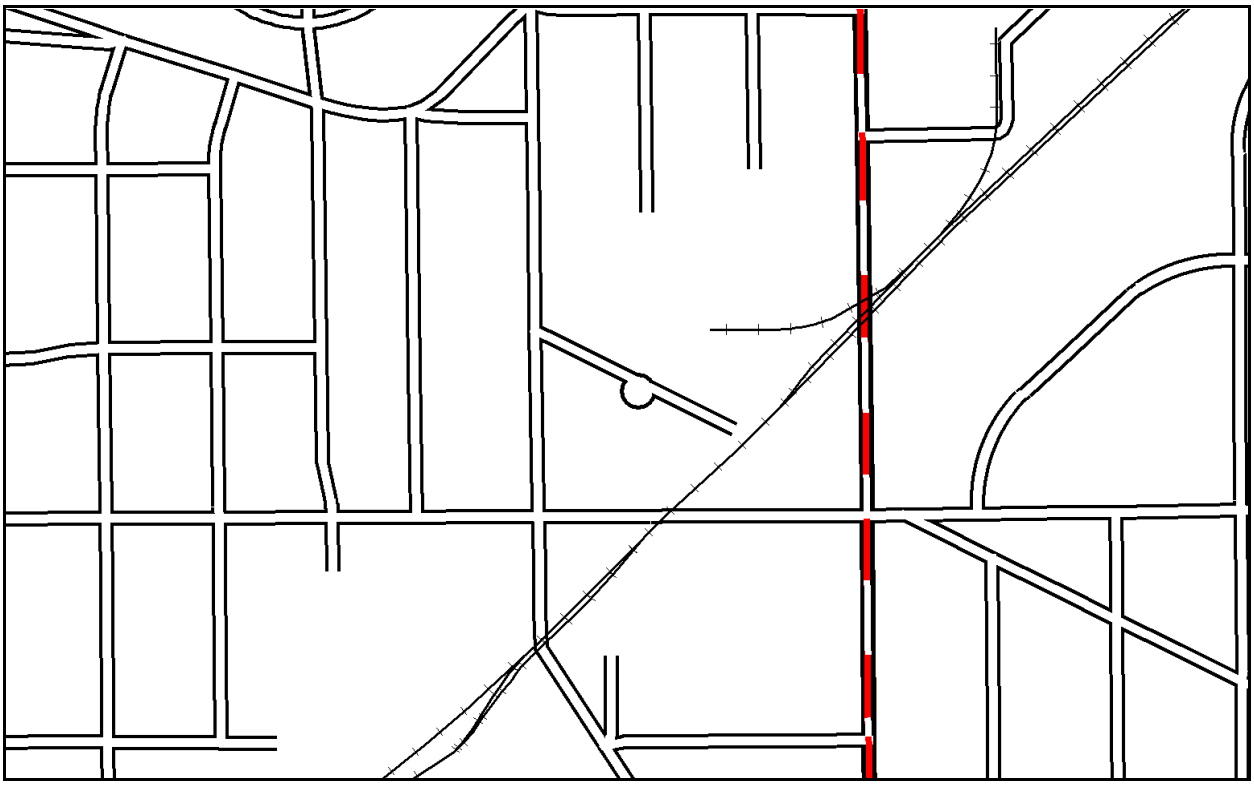


Figure 67 - The same area as the Figure above (zoomed out some), with the streets drawn with casings.

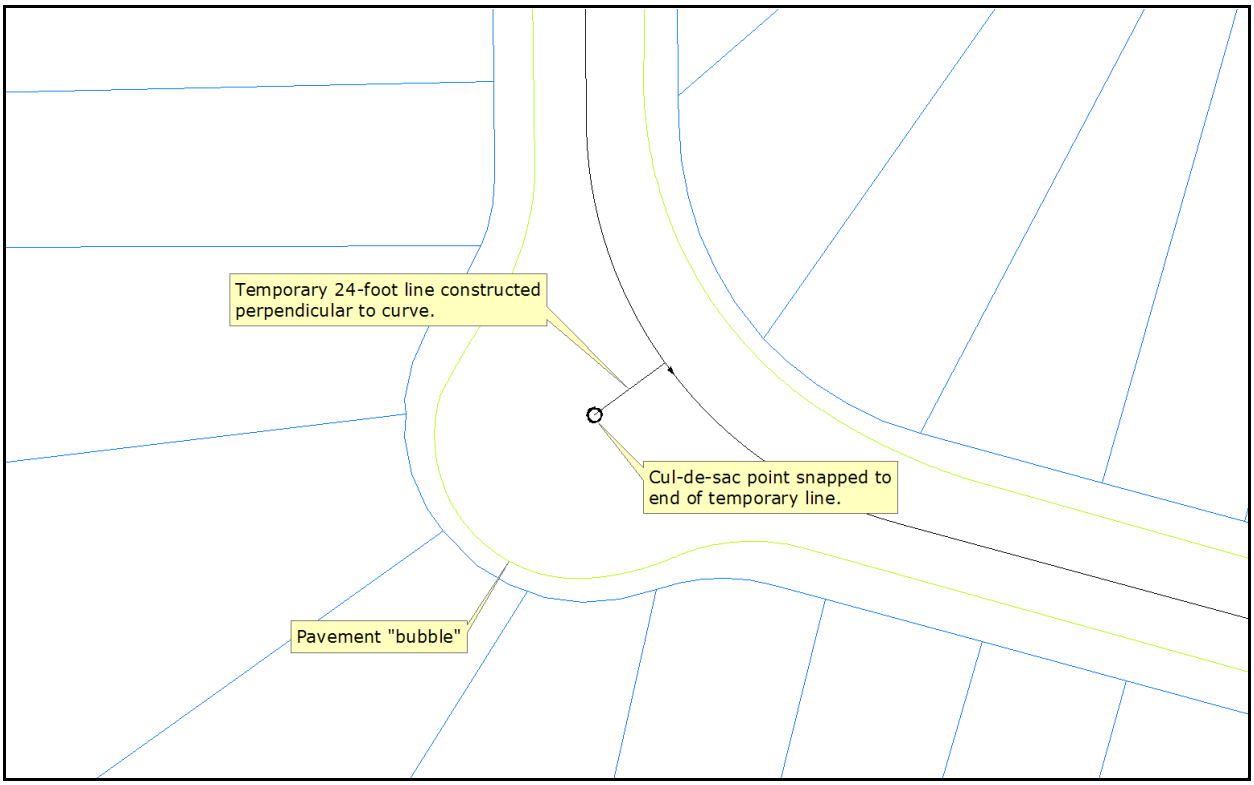


Figure 69 - Example of a cul-de-sac point offset from the centerline on a street curve.

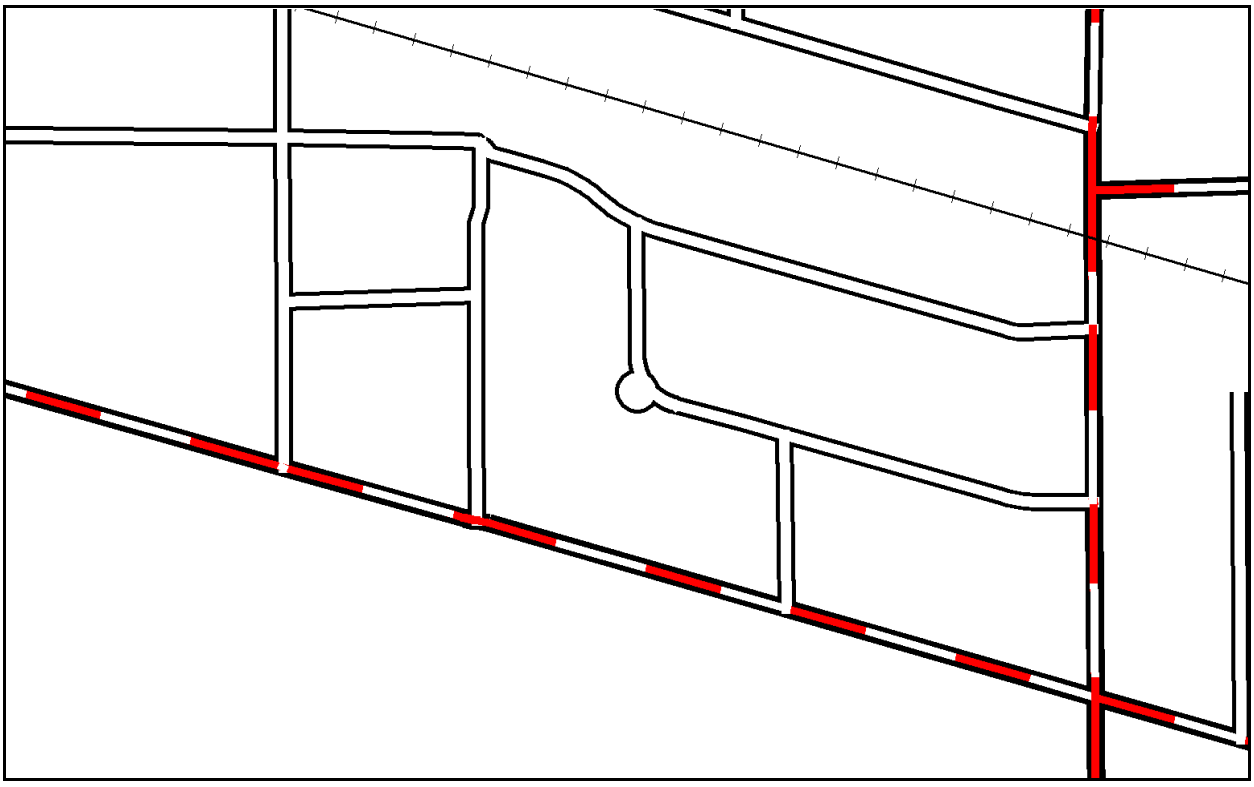


Figure 68 - The same area as the Figure above (zoomed out), when displayed using my street casings Layer File.

The *CuldeSac* layer contains one attribute, OPER_STATUS (the same one that I use on the centerlines), that I use to denote whether or not the cul-de-sacs are constructed. Only four of the possible values are used for the cul-de-sacs, however:

BUILT - The cul-de-sac is on a centerline whose operational status is "BUILT." These points are symbolized by a white circle with a black halo. The white color matches that of built centerlines.

PLATTED - The cul-de-sac is on a centerline whose operational status is "PLATTED." These points are symbolized by a magenta circle with a black halo. The magenta color matches that of platted centerlines.

VACATED - The cul-de-sac is on a centerline whose operational status is "VACATED." These points are symbolized by a yellow circle with a black halo. The yellow color matches that of vacated centerlines.

PLANNED - The right-of-way was platted in order to construct a cul-de-sac, and the street was built, but the cul-de-sac never was, for some reason. These type of cul-de-sac points act as "placeholders," so that I know that the cul-de-sac point wasn't simply missed when I was editing in that area. These points are symbolized by a cyan circle with a black halo. The cyan color matches that of planned centerlines. Note that, however, these cul-de-sacs are *not* located on centerlines whose operational status is "PLANNED," but instead are located on "BUILT" streets. In Figure 70 below.

When the operational status of a street is changed, the attribute of the cul-de-sac point is changed correspondingly.

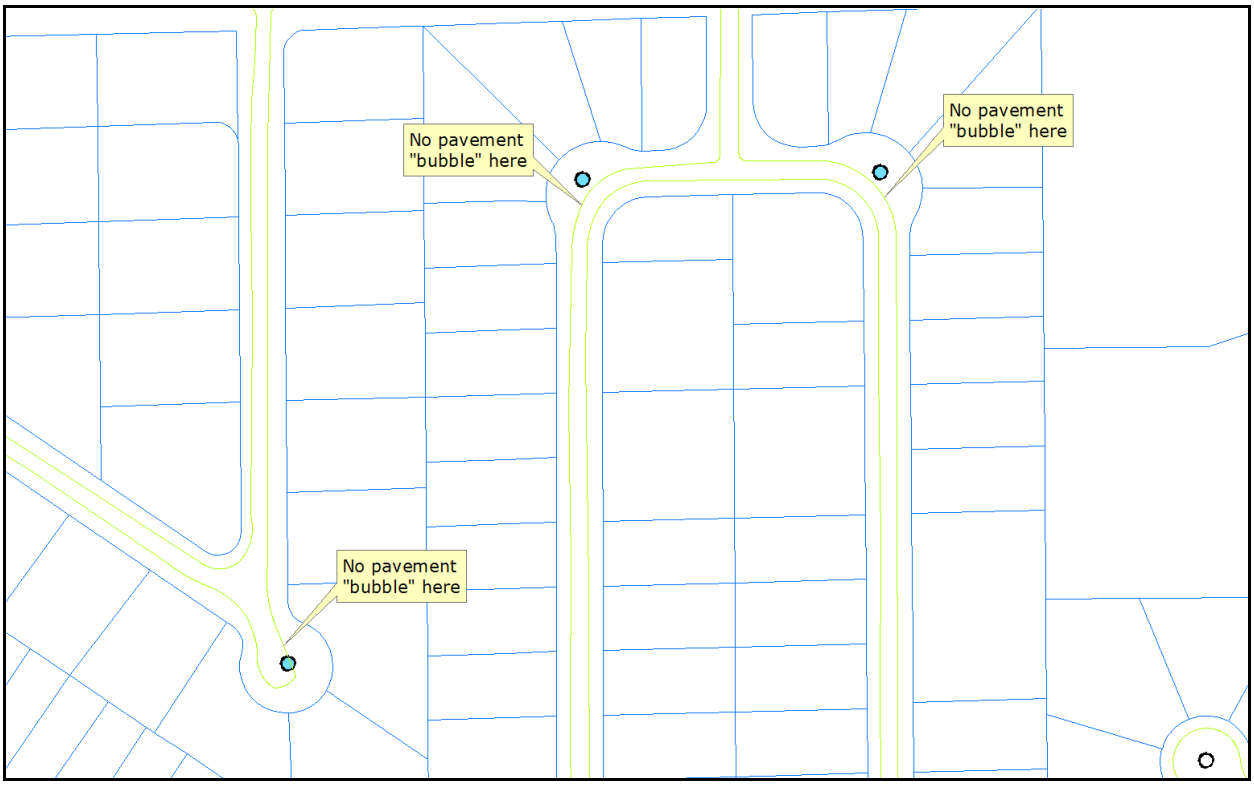


Figure 70 - Some *CuldeSac* points act merely as "placeholders."

The *Seclines* and *SurveySectionCorners* Layers

Back on page 5, I explained how some centerlines follow the survey Section Lines of the Public Land Survey System. In order to do this, I had to create a layer of Section Lines that I could snap the centerlines to. A few years ago I learned that the Office of the Marion County Surveyor had a file that contained very precise GPS coordinates for all the Section Corners in Marion County that they could locate. (The precise coordinates of the corners in the downtown area have been obliterated, due to all the construction that has taken place over the years.) They even had coordinates for many of the quarter-corners and quarter-quarter-corners. Tim Kiste in that office was kind enough to provide me with the file, which I imported into ArcMap. This became the *SurveySectionCorners* layer.

Tim also agreed to provide me with updates to this file, as they became available. He explained that sometimes the Surveyor's Office obtained better information on the coordinates of a particular Section Corner, whereupon they would update the file. Over the years Tim provided me with several updates, before he left employment with the City. No one else in that office has provided me with updates since.

Using the centerlines as a guide, I created the *Seclines* layer by snapping lines between the points in the *SurveySectionCorners* layer, in the areas where I would need to overlay the centerlines. (In the areas where there were no overlapping centerlines, I did not draw the Section Lines.) This layer is shown in Figure 71 on the next page. Obviously, when I received revised coordinates for a corner, not only would the corner have to be moved, but also any Section Lines that were snapped to that corner, as well as any centerlines that were snapped to those Section Lines.

There are places where there are two corners that are very close to each other, oftentimes at intersections. This results in an offset intersection. When I snapped the Section Lines to these corners, I had to be careful which corner I snapped the lines to. In these cases, I relied upon the right-of-way to tell how the Section Lines were aligned. (I knew that the edges of the right-of-way would generally be parallel to the Section Lines.) Even so, there were a few cases where it was impossible to tell which Section Lines belonged with which corners, so it is possible I made a few mistakes.

Figure 72 on the next page shows an example of one of these areas.

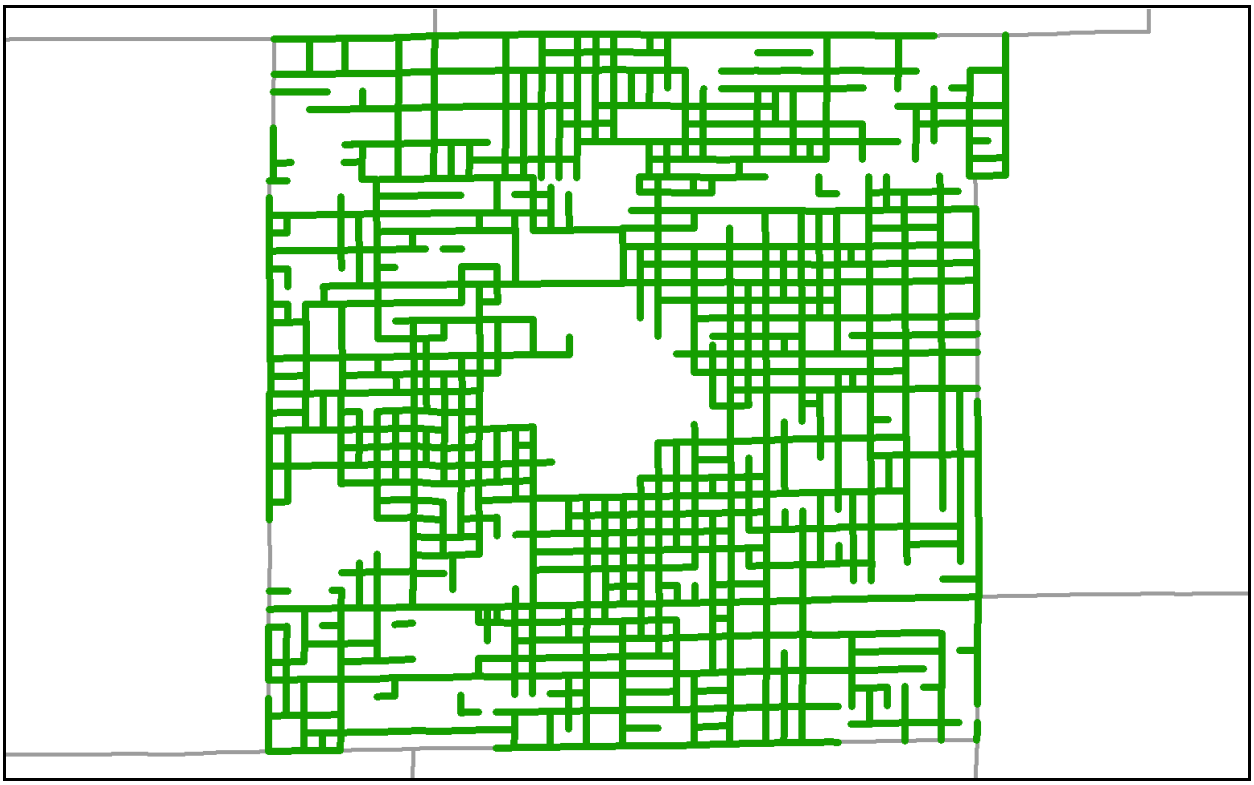


Figure 71 - The *Seclines* layer. The gray lines are county boundaries.

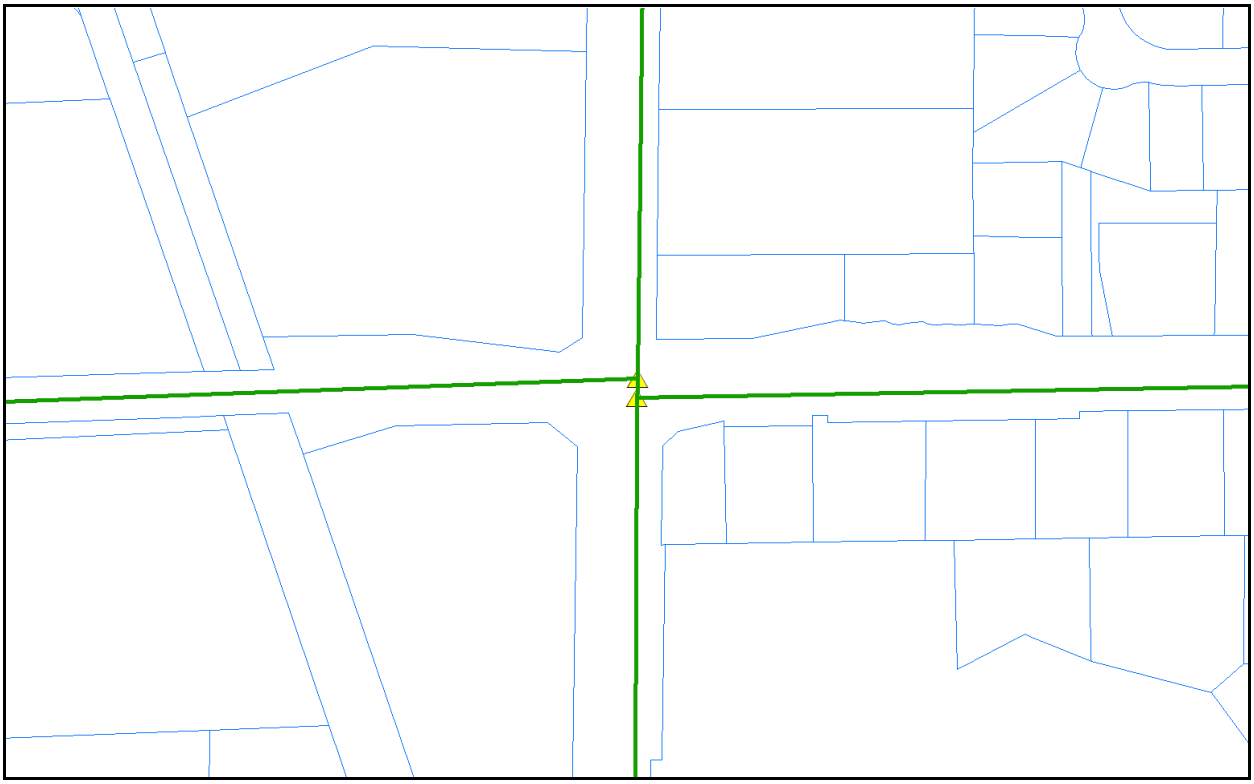


Figure 72 - Example of an intersection that contains two Section Corners close together. The Section Corners are shown as yellow triangles. Note how the edges of the right-of-way are parallel to the Section Lines.

The *PreliminaryAlleys* and *PreliminaryAlleysToo* Layers

PreliminaryAlleys

This layer is now empty, but used to contain preliminary alley centerlines. The alley centerlines were created by extracting the alley right-of-way lines from our Parcels layer, thereby forming casings for the alleys, and then running the ArcInfo CENTERLINE command to generate centerlines from the casings. The layer was created several years ago with the intention that the segments in it would be copied into the master centerline layer (*Streets*) instead of having to digitize them from scratch. The alley centerlines have since been copied over, and so were deleted from this layer. This layer is retained for use as a temporary "scratch" layer, into which I can digitize temporary centerlines that I would rather copy over into the master centerline layer, instead of digitizing them in the master layer. Mostly this happens when I use historical aerial photographs to digitize centerlines of old streets that no longer exist, and I want to avoid the confusion of seeing the centerlines of the new streets overlapping the old streets.

PreliminaryAlleysToo

When the preliminary centerlines for the alleys were created in the *PreliminaryAlleys* layer, they were copied over to this layer before being deleted. When I edit the alley centerlines in the *Streets* layer, these centerlines are displayed behind the ones that are being edited. They are symbolized with a wide, colored line, so that the resulting display simulates all the alley centerlines being highlighted with a highlighter. When I finish editing an alley centerline, such that it is no longer a "preliminary" centerline but a "final" centerline, I delete the preliminary centerline from this layer. In this way, this layer serves as a "placeholder," allowing me to easily see which alley centerlines are still left to be finished. See page 158 for more on editing alleys.

Figure 73 on the next page shows the *PreliminaryAlleysToo* layer as it currently exists. Below that, Figure 74 shows a closeup of some of the preliminary alley centerlines.

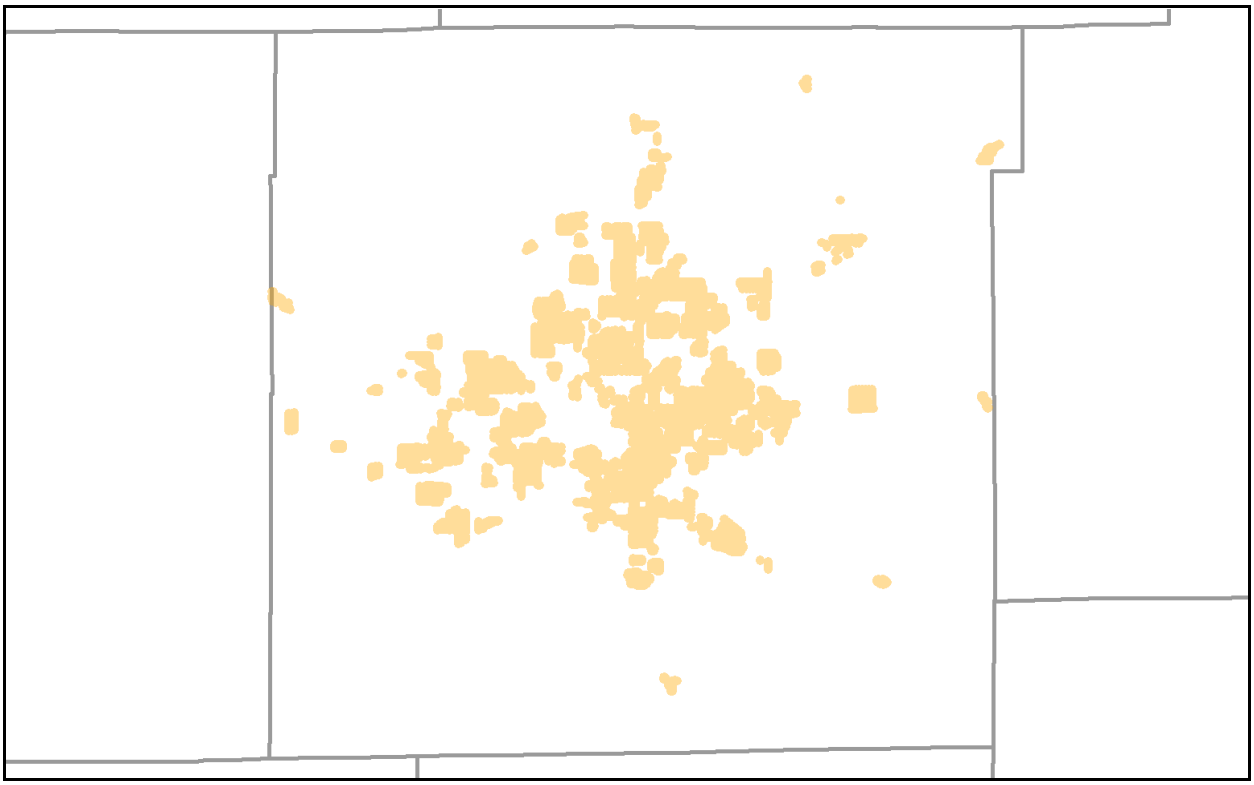


Figure 73 - The *PreliminaryAlleysToo* layer. The gray lines are county boundaries.

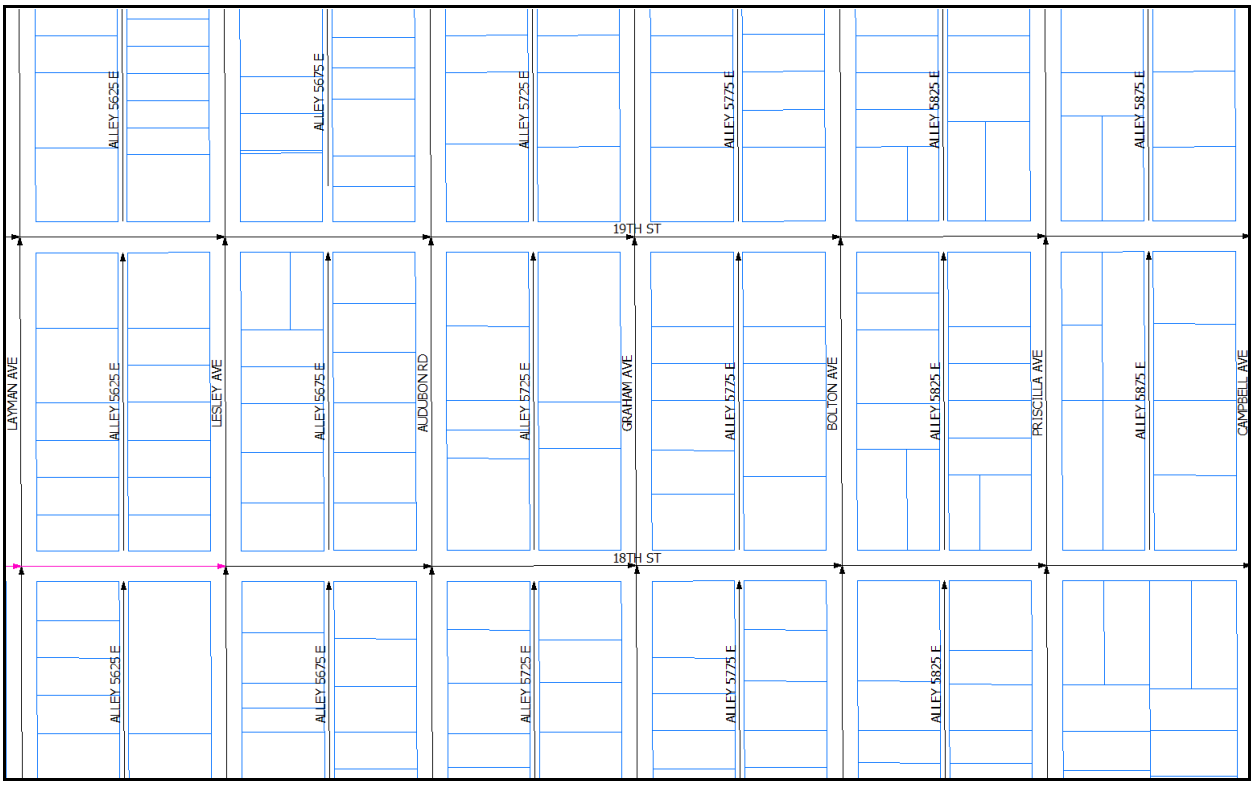


Figure 74 - Preliminary alley centerlines, ready to be connected to the adjoining street centerlines.

There are three layers in the IndyStreets dataset that are empty. Although they are not currently being used, they are being retained in the eventuality they might prove useful someday. They are described in the following sections.

The *TrafficCounts* Layer

Before we made the switch from coverages to geodatabases several years ago, I had a route system on the centerlines called *Tcounts* (Traffic counts). This route system was created mainly in order to be able to make maps of streets with their associated average annual daily traffic (AADT) counts. These traffic counts are provided by the DPW Traffic Engineering Section.

AADT counts are taken periodically on major streets, but not more often than once a year. Each count is considered to be representative of (usually) several contiguous street segments. Typically these limits end at major intersections, but they may be located in mid-block. These portions of a street located between the limits within which representative traffic counts are taken are called "stations," and they don't change very often. (One example of a station is 38th Street from Mitthoefer Road to German Church Road, where one traffic count is used to represent that entire stretch.) Thus, one traffic count is associated (usually) with more than one centerline segment. Therefore, when making a map of the traffic counts, it is desirable to show each count only once, instead of showing the same count for every segment that makes up the appropriate station. This route system allowed you to do that.

The *Tcounts* route system created a one-to-one correspondence between the traffic counts and a spatial feature that could be used to display the counts. Each station is assigned a unique identifier, which is stored in an *xbase* database table. In the *Tcounts* route system, each station is equivalent to one route, and the station identifier is also assigned to the corresponding route. This way, the database containing the traffic counts (which are related to the stations) can be linked to the route system, and the route system displayed on the map, with each route annotated with its corresponding traffic count.

When we switched to geodatabases, it was thought that the *Tcounts* route system would be moved over to the *TrafficCounts* layer, and that I would maintain it that way from now on. However, the move never happened, and so this layer is currently empty. However, I did export the old *Tcounts* route system to a shapefile, which is currently being used to make the traffic count maps. Since the stations and their associated centerline segments don't change very much, there is not much demand to update the layer.

The *StreetClosures* Layer

This layer is another layer, similar to *TrafficCounts*, that was intended to take the place of an old ArcInfo route system, but the switch never materialized.

I used to have a route system called *Stclose* (Street closures) that was created in order to be able to map the locations of closed and restricted streets. (There was talk of putting this information on the City's website.) Each week I received a list of street closures and restrictions from an Internet mailing list published by DPW. I used to create a route for each closure or restriction, based on the information provided in the list. The affected area might consist of one or more segments of a street, or streets, and the limits may not always be at intersections. Each route also contained the other pertinent information for the closure, including the dates of the closure, limits, reason for closure, etc.

I tried to update the street closures route system first thing every Monday morning, since it was a priority to get the updates out in a timely fashion. (The list of closures is distributed late Friday afternoon.) After the route system was updated, I made a shapefile from it, and copied it to a shared area on a server. Chuck Carufel had set up a process that took the shapefile from the shared area and copied it to other servers at 9 AM on Monday. He also had a process that copied it again at the end of the day, in case I didn't get the updates posted by 9 AM.

Since I never heard of any demand for this layer when we moved to the geodatabase format, I quit maintaining it.

The *Trails* Layer

Years ago there was an initiative to create a layer that would show the locations of major trails within Marion County, such that the trail centerlines would share some common attributes with the street centerlines (including address ranges). Consequently, this layer was placed in the same dataset with the street centerlines, which means it would have been my responsibility to maintain them. The initiative never panned out, and thus the layer was never populated.

Subset Layers

All centerline editing is done in one layer, *Streets* (see Figure 75, next page). From this layer, several centerline subset layers are derived and made available to our users. Normally, Chuck Carufel takes care of creating and posting the various subset layers on our servers. These layers are described below.

CNTRLIN - This layer is basically "built streets." It is the most important centerline layer, used by the most people, most often. It is intended to represent all built streets, plus newly-platted streets that will be built soon (mostly new subdivisions), plus platted streets that won't be built, that have developed parcels addressed off them. The net result is that this layer should satisfy most mapping, geocoding, and routing requirements. It is extracted from the *Streets* layer using the following expression:

OPER_STATUS = ("BUILT" or "PLATTED/LOCATOR" or "VACATED/BUILT" or "VACATED/LOCATOR")
or
(OPER_STATUS = "PLATTED" and MAINT_JURIS = ("DEVELOPER" or "PRIVATE"))

See Figure 76 on the next page.

PAPERSTR - This layer, "paper streets," is the exact inverse of the one above. If you take away the CNTRLIN segments from the STREETS layer, this layer is what you end up with. See Figure 77, next page.

INTERSTA - This layer, "interstates," includes the interstates only. It does not include the interchange ramps. This layer, as well as the others described below, makes a good "index" layer for a keymap, or a good background layer for small-scale mapping. It is extracted using the following expression:

STRCLASS = "A" and "OPER_STATUS = "BUILT"

See Figure 78, two pages page.

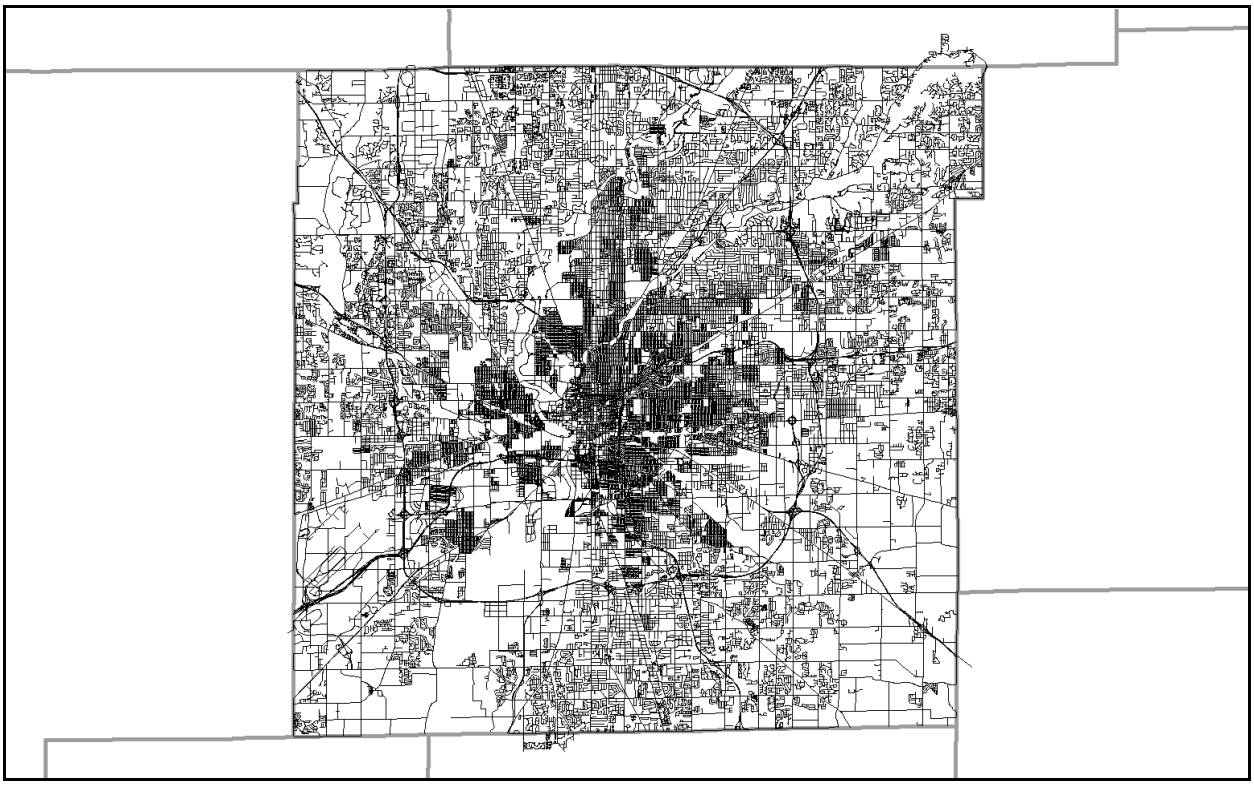


Figure 75 - The *Streets* layer. Notice the streets extending outside the county on the northeast and south sides of the county.

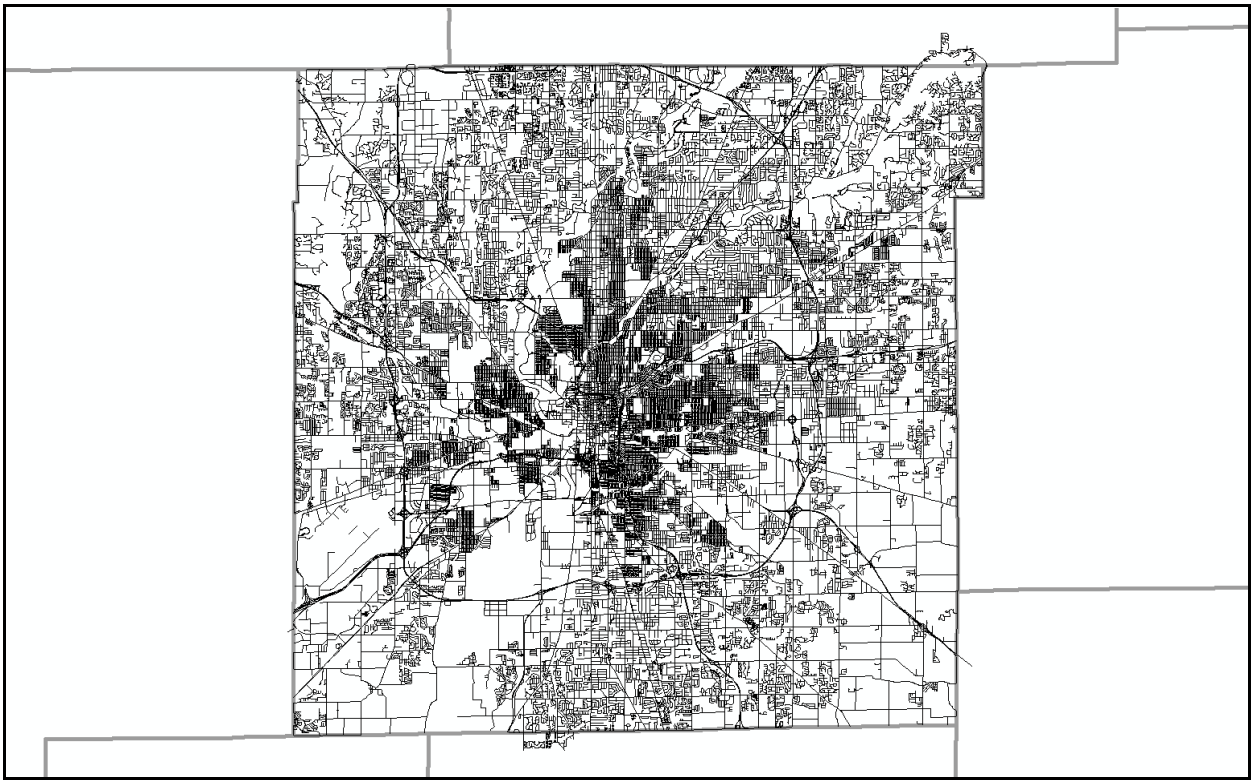


Figure 76 - The *Cntrlin* layer. On first inspection, this layer looks very similar to the *Streets* layer. The *Paperstr* layer (see next Figure) contains those streets that are not in this layer, but are in the *Streets* layer.

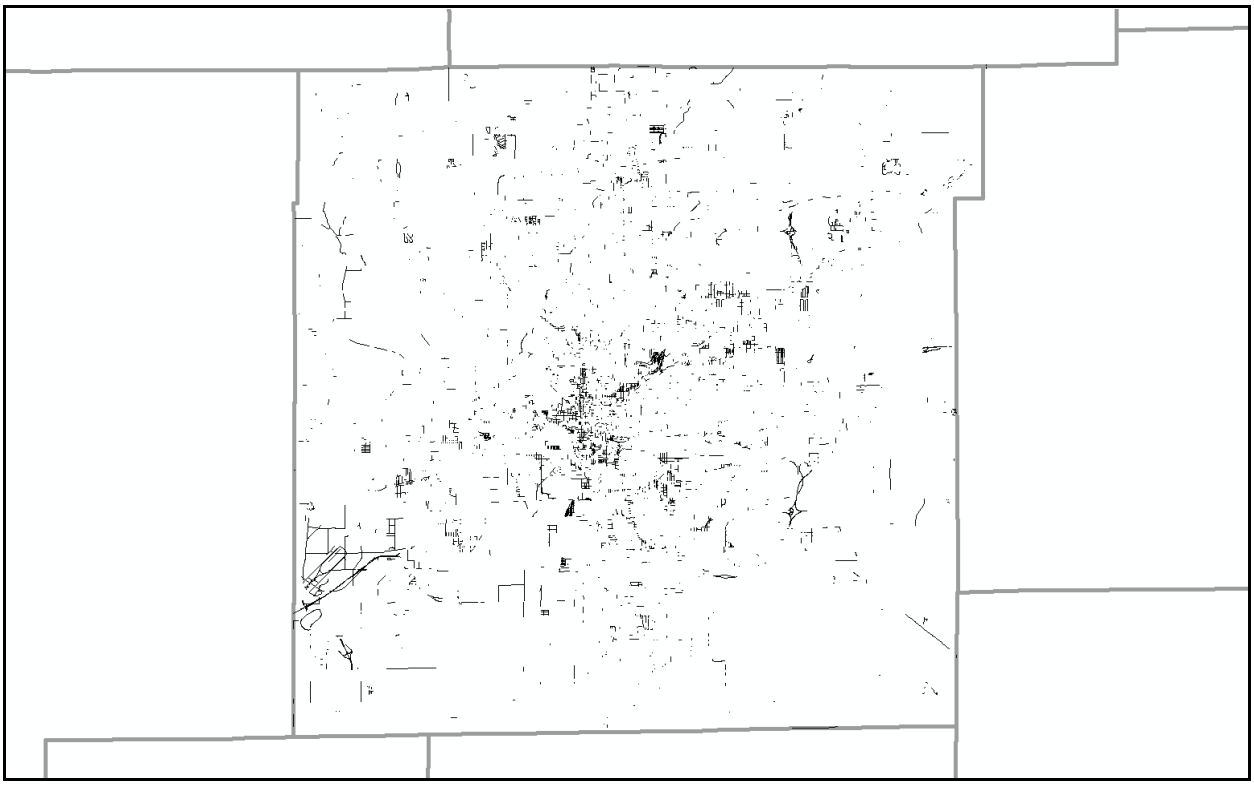


Figure 77 - The *Paperstr* layer.

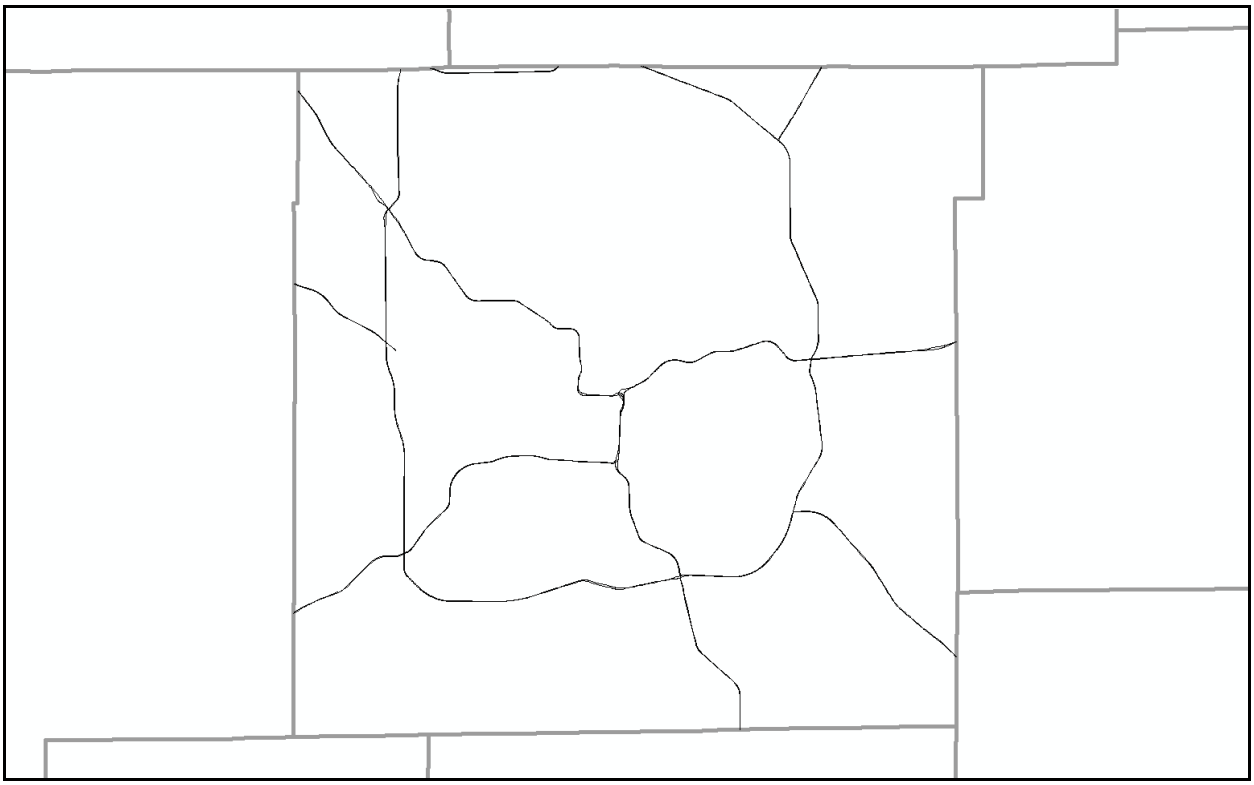


Figure 78 - The *Intersta* layer.

HIGHWAYS - This layer contains interstates and those segments that make up the state highways, including their former alignments through Indianapolis before the interstates were built. It is extracted using the following expression:

STRCLASS = ("A" or "B") and "OPER_STATUS = "BUILT"

See Figure 79, next page.

TFARES - This layer, "thoroughfares," contains the segments comprising the *Official Thoroughfare Plan* of Marion County, published by DMD. It is extracted using the following expression:

TFARE <> "COLLECTOR/LOCAL STREET"

See Figure 80, next page.

MAJORSTR - This layer is "major" streets. (See the definition of major streets in the previous discussion of the STRCLASS attribute, page 54.) It includes the interstates (as well as ramps), current and former highways, thoroughfares, and a few other streets that were added to fill in gaps or empty areas that existed in the *Tfares* layer several years ago. It is extracted using the following expression:

STRCLASS = ("A" or "B" or "C") and "OPER_STATUS = "BUILT"

Note that in recent years, as more streets have been added to the *Thoroughfare Plan*, there is now little difference between this layer and the *Tfares* layer. See Figure 81, two pages over.

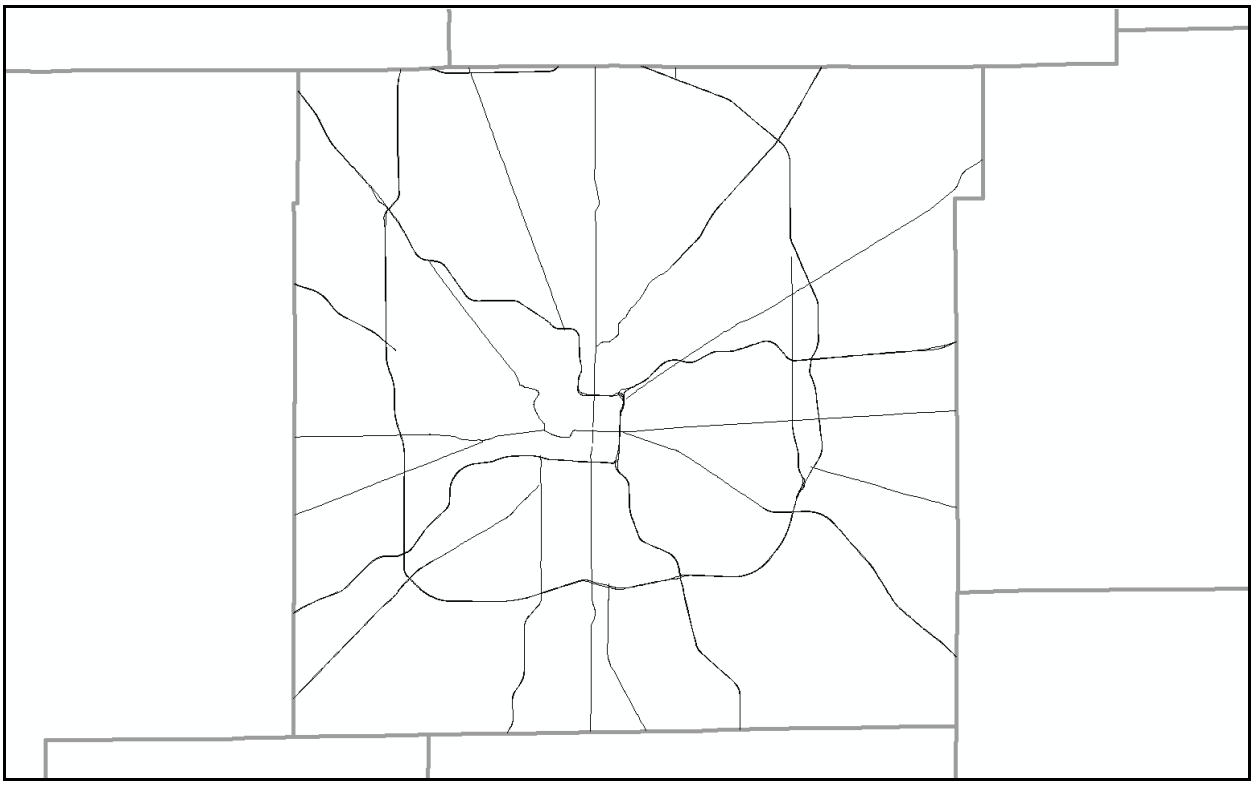


Figure 79 - The *Highways* layer.

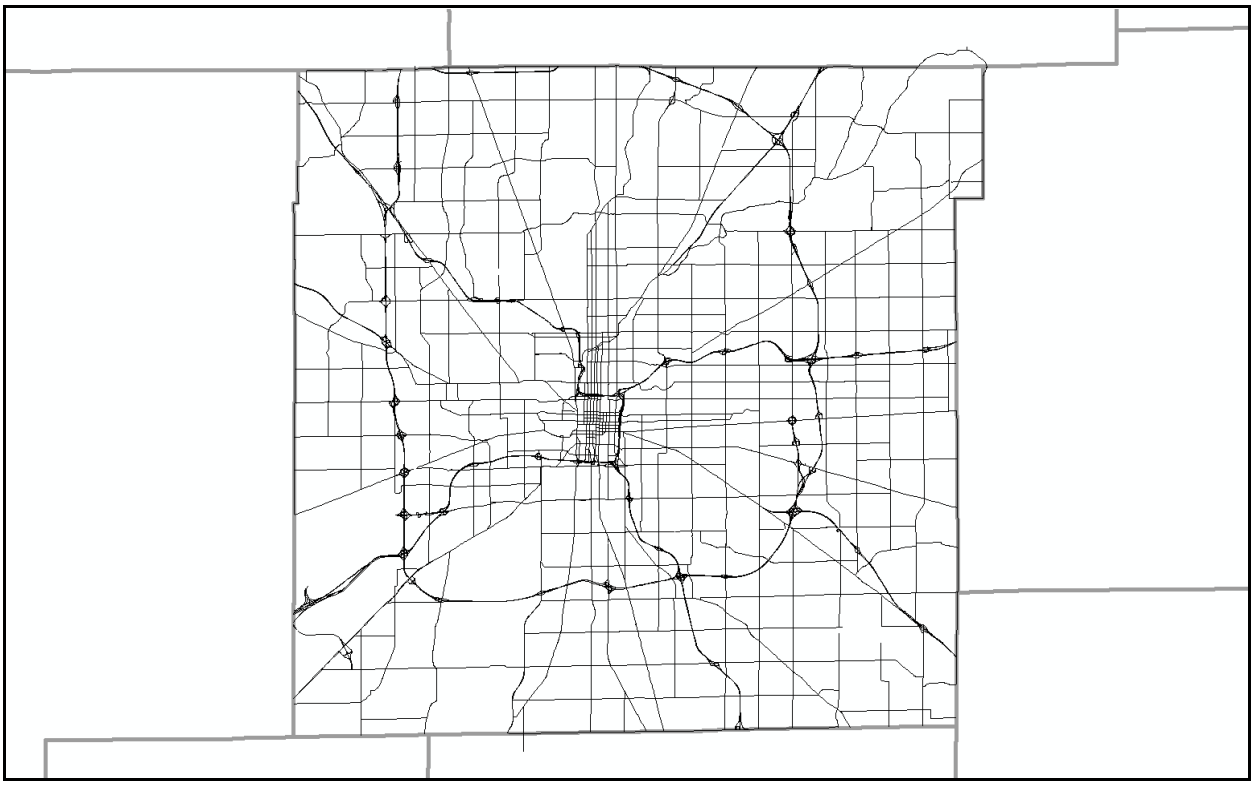


Figure 80 - The *Tfares* layer.

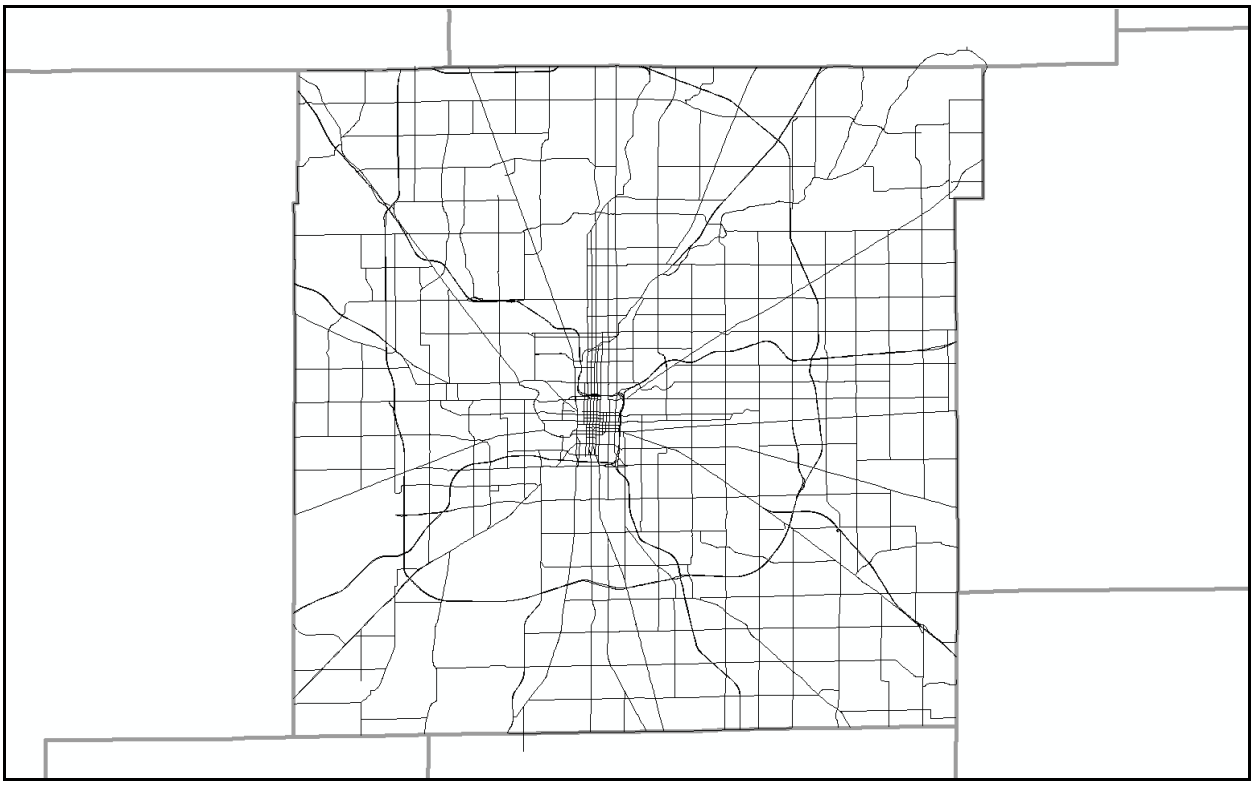


Figure 81 - The *Majorstr* layer.

Old Route Systems

In the past I employed the ArcInfo route feature class to create two route systems on the centerlines that were never transferred over when we made the switch to geodatabases. This is because there is no demand for these layers. These systems were:

OLDTCOUNTS (Old Traffic Counts) - Recall from the discussion of the *TrafficCounts* layer on page 90 that "station" is the term used for a group of contiguous street segments, for which one traffic count is considered representative of all the segments. Although the station limits don't change very often, they do change, due to the ever-changing nature of the street network. I created the OLDTCOUNTS route system in order to be able to save the configurations of outdated traffic count stations, so that someone could look up an old traffic count and see the limits for which the old count applies. This route system also contains the unique identifiers of the stations. (Note: If I remember right, this route system got corrupted once when we were still using the coverage layer format, and it might need to be totally recreated from scratch, if someone ever needs it.)

INTERCON (Interconnected Traffic Signals) - I created this route system in order to be able to make maps of the interconnected traffic signal systems. The source of data used was maps that had been produced by the DPW Traffic Engineering Section. I created the route system basically as a favor to DPW, even though they didn't request it. I created it with the intention of maintaining it, with revisions based on input from DPW, but it was never updated since it was created. DPW did not furnish me any updates, and I did not inquire about any. (Note: I believe this route system also got corrupted, and would need to be reconstructed from scratch.)

Topology

I make use of the geodatabase topology functionality to help insure positional accuracy of the centerlines,

intersections, and cul-de-sacs. The topology is maintained in a feature dataset called *IndyStreets_Topology*, which is the feature class that was created automatically by ArcCatalog, when the geodatabase topology rules were created. Following are the rules that I use.

	Layer	Rule	Layer
1.	Streets	Must Not Overlap	
2.	Streets	Must Be Single Part	
3.	Intersections	Point Must Be Covered By Line	Streets
4.	Intersections	Must Be Covered By Endpoint Of	Streets
5.	Seclines	Must Not Overlap	
6.	Seclines	Must Be Covered By Feature Class	Streets
7.	Cul-de-sac	Must Be Covered By Endpoint Of	Streets

Ranks:

Seclines	1
Streets	2
Cul-de-sac	3
Intersections	3

Rule one is self-explanatory. It does have some exceptions, as explained on page 21.

Rule two is also self-explanatory. It has no exceptions.

The purpose of rule three is to ensure that all intersections are snapped to centerlines. The only exceptions are certain roundabout intersections, as shown in Figure 59.

Rule four does have some exceptions as well. Besides the ones included in the previous rule, this also includes overpass intersections.

Rule five is self-explanatory, and has no exceptions.

Rule six has some exceptions, where streets were never built along a section line. See Figure 82 on the next page.

Rule seven has a few exceptions as well, where pavement bubbles occur in mid-block. See Figure 65.

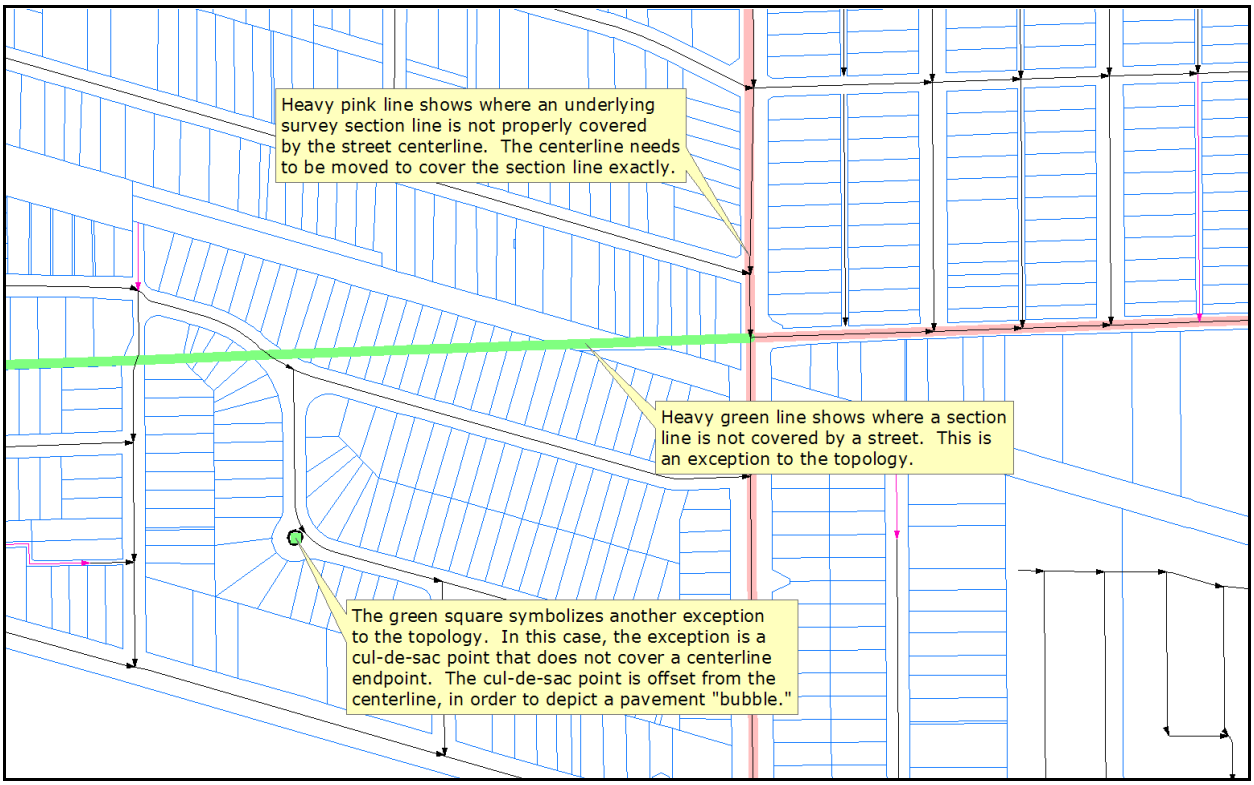


Figure 82 - Not all section lines have streets built along them. These areas are exceptions to Topology Rule #6.

METHODOLOGY

In this section I get into how I actually perform my work.

Using a Version

All my editing is done using a version. This is so I can save my work (to the version), without having to worry about having to wait on other people saving and posting their edits at the same time. Before we adopted versions as the preferred methodology for doing edits, we would frequently have to wait to save our changes while someone else was saving and/or reconciling and posting their changes. (Two people cannot save their work at the same time to the same version.) Sometimes we had to wait as much as 15 to 30 minutes for the other person's work to post before we could save our edits.

The version I use for centerline editing is for my exclusive use. Created from the DEFAULT version, It is called MBRADLEY and it is created with "Protected" permissions, meaning other people can see it but can't edit it. I try to reconcile and post my changes at least once a week, usually on Friday afternoons. However, sometimes I need to post more frequently. For example, whenever I enter centerlines for a new subdivision, I need to post my work right after I finish entering the centerlines. This is so Brian Schneider can assign addresses to the adjoining parcels. (Recall from the discussion about the Master Address Database on page 2 that parcel addresses cannot exist without first existing on the centerlines.) Entering new subdivision centerlines and parcels into our GIS is of the highest priority.

Sometimes our database administrators will want to perform a full compression on our GIS database. In order to do this, all versions must be removed first. Typically, the compression is done over the weekend. When a compression is planned, GIS users like myself will be requested to delete their versions before leaving on Friday, and then recreate them on Monday morning before editing.

General Editing Policies

- I use a snapping tolerance of 16 pixels.

- I almost always have the vertices displayed while editing the centerlines. This is because so many times the locations of the vertices are dependent upon the geometry of the other layers, that it is crucial to ensure the vertices are located in the right position.

- I try not to introduce superfluous vertices into the centerlines. Sometimes we call these "pseudo nodes." (This term originated back in the days of ArcInfo coverages.) Pseudo nodes are those vertices on a centerline where the direction of the line does not change at the vertex. Thus these vertices are redundant.

  There are two reasons for not introducing pseudo nodes into the database. One is to try and keep the database as simple as possible. Granted, in this day of massive hard drives, this may not be a big consideration. The other reason is that if I come upon an area that I edited previously, it is not always obvious from looking at the centerlines (with the vertices turned on) if a vertex is superfluous or not. For example, consider Figure 83 on the next page.

  Is the highlighted vertex in the Figure superfluous? Maybe yes, and maybe no. It depends on whether or not the alley centerlines to the north and south of the cross street are in perfect alignment. If they are, the vertex is superfluous. Without close inspection, it is not obvious whether this is the case or not. (Read how alleys are maintained on page 102.) Assuming this vertex is redundant, it behooves me to eliminate it, so that if I ever come back to this area, I don't have to wonder if the vertex is needed or not.

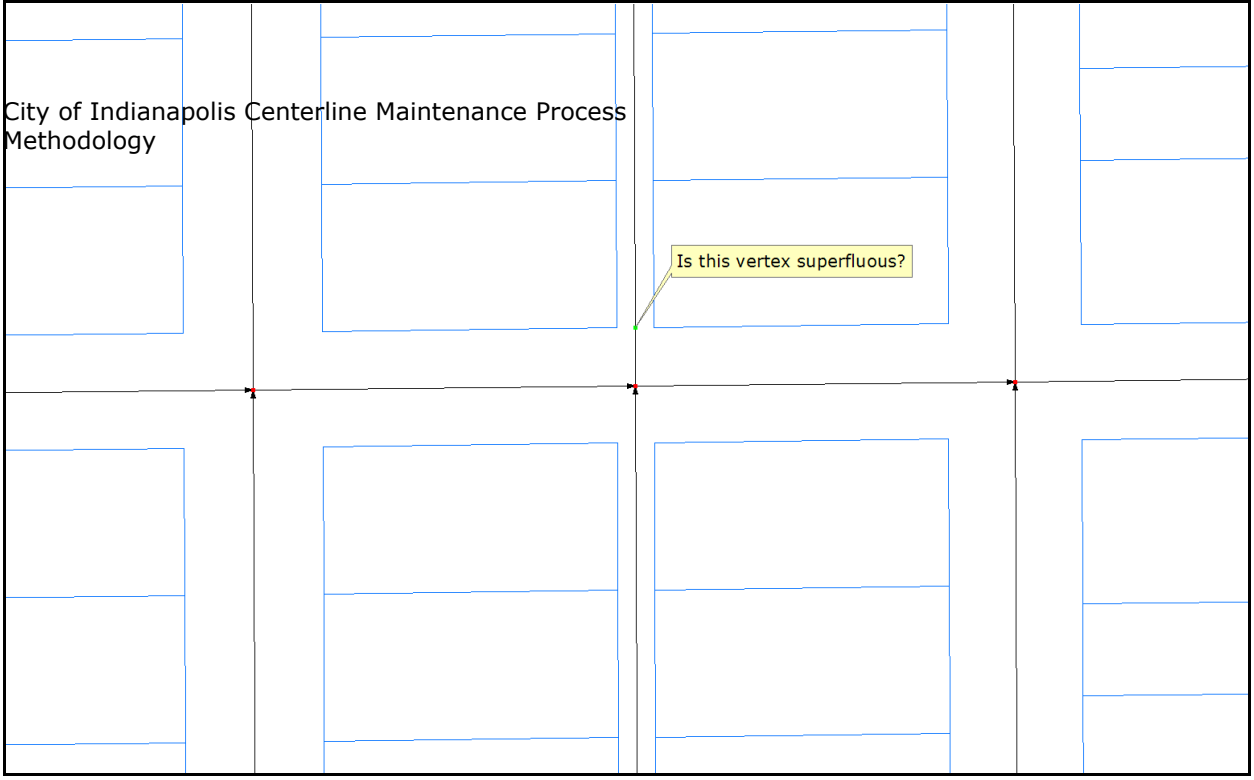


Figure 83 - Sometimes extra vertices may be introduced during the course of editing.

Extra vertices are commonly introduced when temporary centerlines are sketched in order to find the proper geometric locations for vertices or endpoints of permanent centerlines. Another case when pseudo nodes are created is when an existing centerline is extended. (A pseudo node is automatically inserted at the original end of the centerline.) Whatever the origin of these unwanted vertices, I try to eliminate them as much as possible.

- My default snapping setting is to have vertex, edge, and end snapping enabled for the *Streets*, *StreetsOOC*, and *Parcels* layers, and no snapping turned on for any other layers. When snapping to the *Parcels* layer, if I want to make sure I snap to a vertex, I hold down the <V> key in order to display the vertices in that layer.

- When I'm done working in an area, I always validate the topology of the area (unless I forget!), and make sure everything involved in the topology is kosher. I do this by pressing the Validate Topology in Current Extent button. I never use the Validate Topology in Specified Area or Validate Entire Topology buttons. I have my topology settings set to display dirty areas.

- I try to remember to save my work frequently, say, every ten minutes. ArcMap does still crash occasionally!

- There is a little utility called "Advanced ArcMap Settings" that ships with ArcMap. (See Figure 84, next page.) In a typical default installation, it is located in the C:\Program Files\ArcGIS\Utilities folder. There is one setting in this utility that is important to me. On the Editor tab, there is a checkbox labeled *Initialize default values on subtype change*. It is important that this value not be checked. Otherwise, when I change the subtype value of a centerline, all the existing attribute values that have default values assigned to them for each subtype, will be replaced with their new subtype default value. This is NOT what I want.

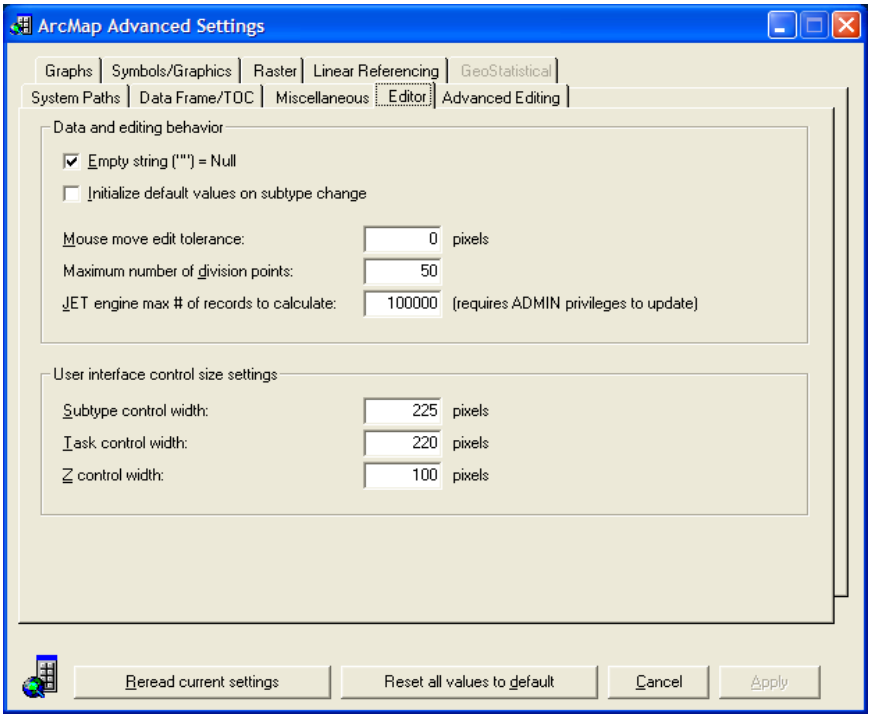


Figure 84 - Advanced ArcMap Settings utility.

New Streets

Most centerlines for new subdivisions and commercial developments are provided by Brian Schneider of DMD. The only exception is centerlines that Brian missed, that are apparent from the parcels and/or aerial photos. (This doesn't happen very often.) Brian also provides centerlines for apartments and other private streets, but in some cases, I augment these with additional centerlines (side streets) apparent from the aerial photos.

When Brian receives a new plat or apartment plan from Doug Lynch, he digitizes the centerlines using COGO tools. The result is a MicroStation Design (.dgn) file, which Brian then sends to me via e-mail. On the average, I receive one from Brian about every other week. The .dgn file actually contains six separate layers. I make use of two of them, the Polyline layer and the Annotation layer. The Polyline layer contains the geometry, and the Annotation layer contains the street names and address ranges. (See Figure 128.) If the streets are private, Brian will indicate that in the e-mail; otherwise, I am to assume they will be public.

I save the new .dgn files from Brian in a directory on a server that is reserved for my files. It is located at H:\GIS\NewCLs. When Brian sends me the files, he names them after the name of the development, but when I save them, I rename them using the format "clXX_XXXXXX." In this convention, "cl" stands for centerline, the first two "x'"s are a consecutive number from "01" to "99," with leading zeros, and the last six "x'"s are the date I received the centerline file. (first two digits are the last two digits of the year, followed by the month (with leading zeros), followed by the date (with leading zeros). As an example, "cl07_041110"would be the seventh set of centerlines received from Brian, and they would have been received on November 10th, 2004. This numbering scheme allows the .dgn files to sort nicely. Once I get 99 sets of centerlines in the directory, I archive them onto a CD, erase them, and start the process all over.

When I am ready to input the new centerlines, I paste them into my map, clean up the geometry and topology, and populate the attributes. When this is done, as mentioned before, I reconcile and post the changes. Inclusion of new centerlines in the street network is usually the highest priority of all the centerline projects I work on. Thus I temporarily suspend any other projects until the new centerlines have been added.

Existing Streets

Existing centerlines are constantly evolving. The geometry and attributes need to be updated as the characteristics of streets change. New subdivisions, capital improvement projects, private developments, vacations and abandonments, street closures, notices of errors, and other factors all combine to make this a never-ending process. In addition, Brian Schneider sometimes moves all the parcels of an entire subdivision at one time, for one reason or another. When this happens, the centerlines must be moved as well. (The parcel locations take precedence over the centerlines.) In addition, the alley centerlines need to be completed.

I try to make changes to the centerlines as soon as I become aware of the changes to their streets, from whatever source. Some projects, such as adding alleys and historical streets, can be done whenever there is free time. There are always things that can be done to make the centerlines better. (For some examples, see the list of goals for the centerlines at the end of this document.)

Usually, changes come about randomly, but occasionally, I will do a systematic sweep of the entire county while performing some type of update. An example would be when we receive a new set of aerial photos (new photos are taken every year), I scan the entire county, looking for streets whose status needs to be changed from "Platted" to "Built." However, due to the length of time involved in a countywide sweep, this process always gets interrupted when something more pressing needs to be done. That is why I sometimes create tables or lists to help me keep track of what areas I've finished. Sometimes, I use the DMD Basemap boundaries as my area of reference.

There are times when I want to revise the shape of a bunch of centerlines in an area. For example, there are still lots of areas where the centerlines don't follow the center of the right-of-way (or center of pavement, if they are private streets). In cases like this, I will usually temporarily move the whole bunch of existing centerlines out of the way, digitize the new centerlines in the correct location, transfer the attributes from the old centerlines to the new ones, and then delete the original centerlines.

Alleys

As mentioned on page 88, preliminary alley centerlines have already been created for all the alleys. The attributes on these preliminary alleys have already been populated, including their names. In the interest of speed, when these preliminary centerlines were populated, they were all assigned an operational status of "BUILT."

The task now is to revisit the alley centerlines, and clean them up. This includes extending them to their adjoining street, and changing the OPER_STATUS value to what it should be. (The most recent aerial photos are used for this, sometimes in conjunction with Pictometry photos — see next section.) Sometimes only part of an alley is traversable, which means the alley centerline needs to be split in order to show this. The whole process of cleaning up the alleys involves a considerable amount of work, and will probably take years to complete. See page 158 for more details on how this is done.

I like to work on alleys in between my other tasks, as it gives me some variety from the other monotonous tasks I do. And I like to work in different areas of the county at different times, also for variety. There is no deadline to have all the alleys done, which is a good thing! Typically, I will choose a rectangular area of the county to work in, and finish all the alley centerlines within that area at one time. To find an area to work in, all I have to do is zoom out to the extent of the entire county, and turn on the *PreliminaryAlleysToo* layer, which shows all the remaining alley centerlines that need to be cleaned up.

Pictometry

Pictometry is a company that provides digital, oblique aerial imagery. For the last few years the City of Indianapolis has licensed from Pictometry their oblique photos of Marion County, which have been made available to users throughout the enterprise, including GIS. I have found the oblique photos to be helpful

in my centerline work, mostly to help determine which portions of alleys are traversable and which are not.

Pictometry provides an extension for ArcMap that enables you to access the photos from there, without having to use their own viewing software. Unfortunately, the Pictometry toolbar doesn't work for me. Whenever I click on a button to display a photo, nothing happens. I think it is probably a memory issue, related to the fact that I have so many toolbars and layers in my ArcMap document. Therefore, I have resorted to relying on the Microsoft Virtual Earth website (http://maps.live.com/) to view the oblique photos! These are the same photos that we have access to by way of our license, although the website provides fewer angles and elevations. However, sometimes the photos on the website are newer than the ones we are licensed to use!

Setting Up a Typical Editing Session

In this section I outline the exact system setup I use to edit the centerlines.

I have two flat-screen monitors at my workstation. The one on my left is a 27" monitor; the one on my right is an 18" monitor. I use them both while working on the centerlines. I display my ArcMap window on the larger monitor, and on the smaller one I display various related windows. Figures 85 and 86 on the next two pages show how my setup looks.

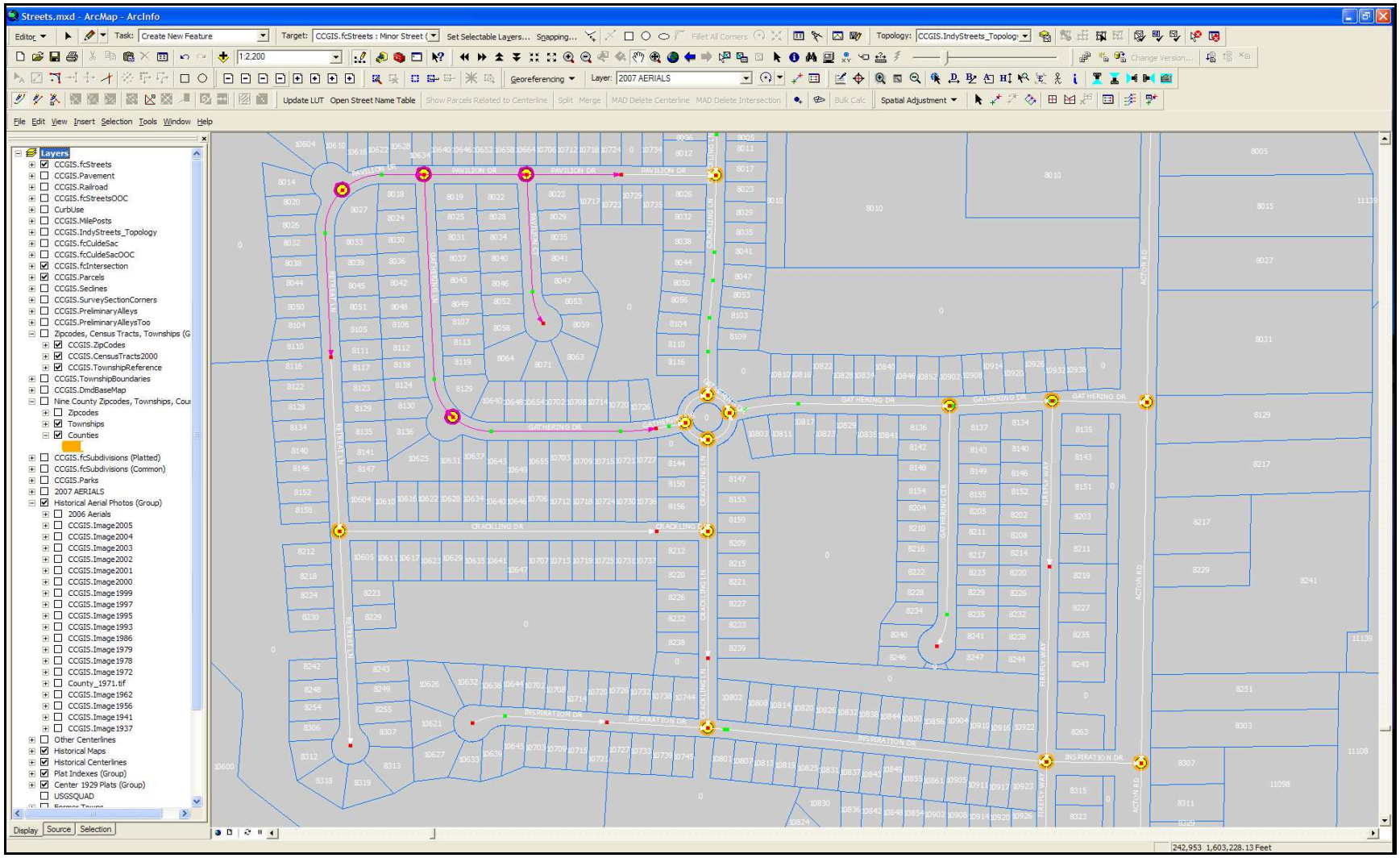


Figure 85 - A typical display on my left monitor.

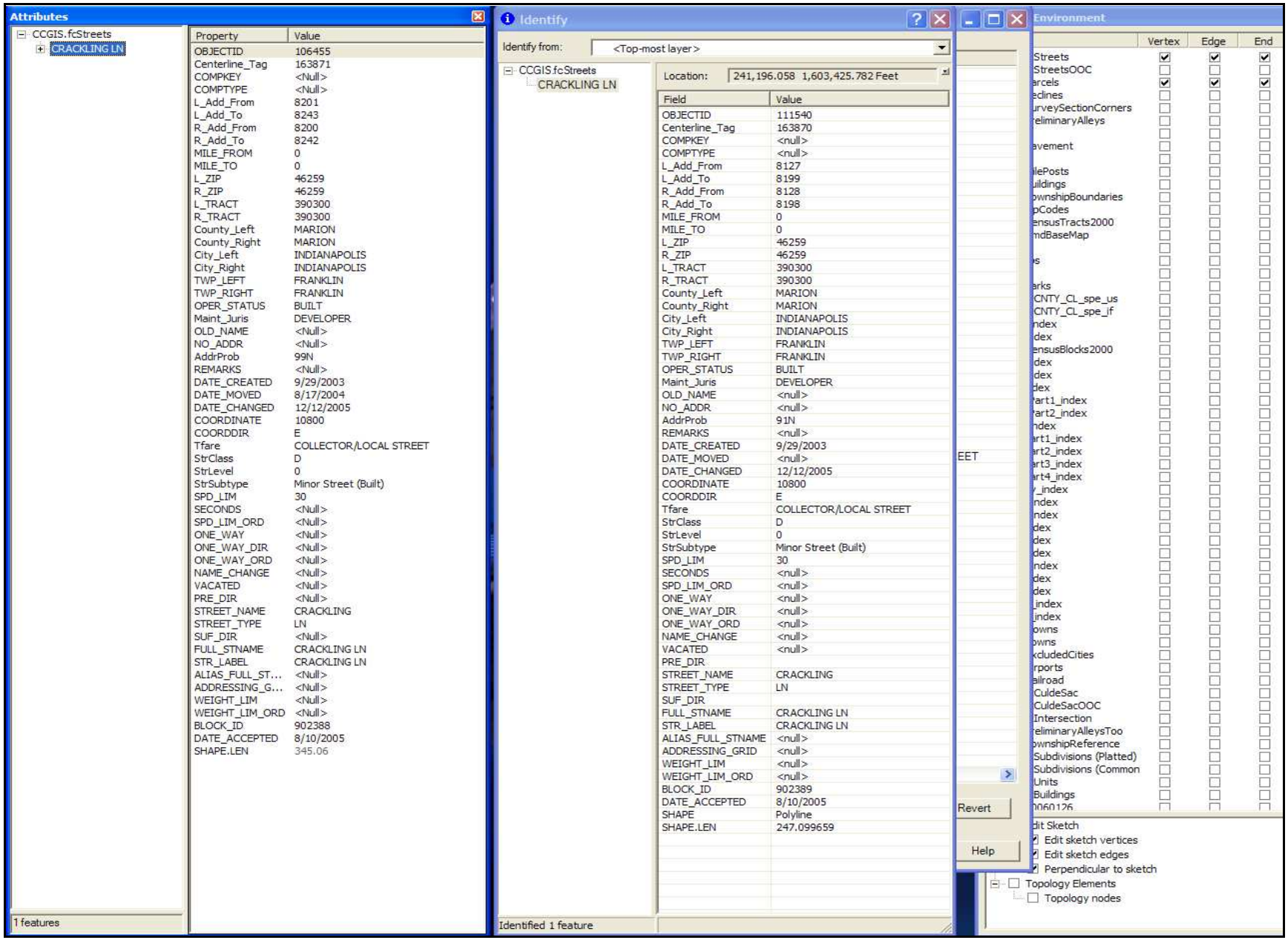


Figure 86 - A typical display on my right monitor.

The Left Monitor Display

The setup you see in Figure 85 is of course saved in a Map Document; I call it simply *Streets*. It is saved in D:\GIS\Map Documents on my harddrive, along with my other Map Documents.

The toolbars at the top of the screen are a mixture of out-of-the-box toolbars and some custom toolbars. When we first acquired ArcGIS with version 8.0, there was a lot of functionality that was not built into ArcMap, that I wanted and needed to be able to edit the centerlines. Basically, I needed to be able to replicate the functionality that I had available to me when our centerlines were still in the coverage format, and I had a whole slew of AML's at my disposal. Consequently, I went to the *ArcObjects Online* section of the ESRI website, and downloaded several custom tools to add to the out-of-the-box tools that ArcMap came with. I installed the tools following the included instructions. The functionality provided by some of these tools has since been added to the generic ArcMap package, but because I was already familiar with the custom tools I had downloaded, I decided to continue to use the custom tools, even though they may have been replaced with out-of-the-box tools.

Just to prove to myself that I could do a little *ArcObjects* programming, I also created myself a custom zoom tool, so that I can zoom the display in and out at predetermined intervals.

The last custom tool that I use is one developed by a consultant specifically for us, as an interface between our geodatabase (ESRI's ArcSDE database) and our Master Address Database.

Here is a list of the out-of-the-box toolbars that I have turned on:

Advanced Editing
Editor
Georeferencing
Main Menu
Map Cache
Spatial Adjustment
Standard
Tools
Topology
Versioning

Tools I Downloaded

Line Edit Toolbar - This was the first and most important of the tools I installed. Below is what this toolbar looks like.

Here are what each of these buttons do, numbered left to right:

1. Toggles the display of the vertices and endpoints of the centerlines. Doesn't matter if any centerlines are selected or not.
2. Toggles the display of arrows on the centerlines, in order to show their digitized direction. I don't use this tool, as I already have the centerlines symbolized to show their direction.
3. Toggles the display of dangles of *selected* centerlines.
4. Splits one or more selected centerlines at their midpoint.
5. Divides selected centerlines into a selected number of equal-length pieces, or pieces of a given length.
6. Splits a selected centerline at a specified length from its end, or at a specified percentage of its length from its end.
7. Modifies the length of selected centerlines. I don't use this tool.
8. Removes dangles from a selected set of centerlines. I don't use this tool.

9. [See the "Split at Intersection (1)" tool described below.]
10. [See the "Split at Intersection (2)" tool described below.]
11. [See the "Flip" tool described below.]
12. [See the "Extend Line to Feature" tool described below.]
13. Creates a polygon from a selected set of centerlines that close to form a polygon. I don't use this tool when editing centerlines.
14. Replaces an edit sketch with a formerly selected set of centerlines that form a chain. I don't use this tool.

Split At Intersection (1) - This tool consists of a single button. I installed it inside the "Line Edit Toolbar" toolbar, because of its related functionality. It is shown as button #9 in the illustration for that toolbar. This tool splits selected centerlines where they cross. If a centerline is intersected by more than one centerline, it will only be split at the first intersection found.

This tool has been superceded by the "Planarize Lines" tool on the standard Topology toolbar. However, I have found a difference between the two tools that I use to my advantage. Oftentimes I want to split the intersection of centerlines coming into an intersection with a circle that surrounds the intersection. This happens in cases where I need to warp the centerlines so they intersect at a logical intersection. (See Figure 14.) I have discovered that when I intersect the street centerlines with the circle using this tool, it splits the street centerlines, but doesn't split the circle. This is precisely what I want, because I don't care if the circle is split; I only need to split the street centerlines. After the centerlines are split, I can easily select the entire circle and delete it. If I use the "Planarize Lines" tool, it will split not only the street centerlines, but the circle as well, leaving me with four pieces of the circle to select and delete.

Split At Intersection (2) - This tool consists of a single button. I installed it inside the "Line Edit Toolbar" toolbar, because of its related functionality. It is shown as button #10 in the illustration for that toolbar. This tool will split selected centerlines where they are crossed by another centerline that you touch with this tool. ArcMap offers similar functionality with a different tool, but you must draw an edit sketch to mark the point of intersection to split a line at; with this tool, if an existing centerline already intersects where you want to split the selected line, you don't have to draw an edit sketch.

Flip - This tool consists of a single button. I installed it inside the "Line Edit Toolbar" toolbar, because of its related functionality. It is shown as button #11 in the illustration for that toolbar. This tool flips the direction of selected centerlines.

Extend Line to Feature - This tool consists of a single button. I installed it inside the "Line Edit Toolbar" toolbar, because of its related functionality. It is shown as button #12 in the illustration for that toolbar. This tool extends a selected centerline to the first centerline it finds in the direction that you click with this tool. I don't use this tool anymore because ESRI provides similar functionality with one of ArcMap's standard tools.

Custom Fillet Tool - This tool consists of two buttons. Below is what it looks like.

The first button allows you to create a fillet of a given radius between two intersecting centerlines. The radius is in map units.

The second button will fillet all corners of one selected centerline with a given radius. The

radius is in map units.

See *Using the Custom Fillet Tool* on page 148.

Auto Shapes - This tool consists of three buttons. Below is what it looks like.

The first two buttons have been superceded by similar tools in the generic "Advanced Editing" toolbar. However, the third tool allows you to create an ellipse. Very few streets are actually shaped like an ellipse (or part of one), but we do have a few, and I have used this tool for them.

The tool works by dragging out a rectangular shape on the map, and it inserts an ellipse that fits within that bounding rectangle. I usually end up scaling and rotating the ellipse to get the shape of the curve to exactly match the geometry of the street I'm trying to reproduce. Of course, I usually don't need the entire ellipse for a street curve, so I have to trim off the unnecessary portion.

Smooth Tool Control - This tool consists of a slider, and looks like this.

This tool "smooths" a centerline with corners, by replacing the centerline with one consisting of Bezier curves. The farther you slide the slider to the right, the bigger the curves used. This tool provides more "granularity" than the Smooth tool on the ESRI Advanced Editing toolbar. See Figures 87 through 89 on the next two pages for examples of the output.

I use this tool infrequently, but occasionally it has its uses, especially when digitizing a private street or dirt road with many random curves. The tool is very touchy — just the smallest amount of smoothing accomplishes want I need.

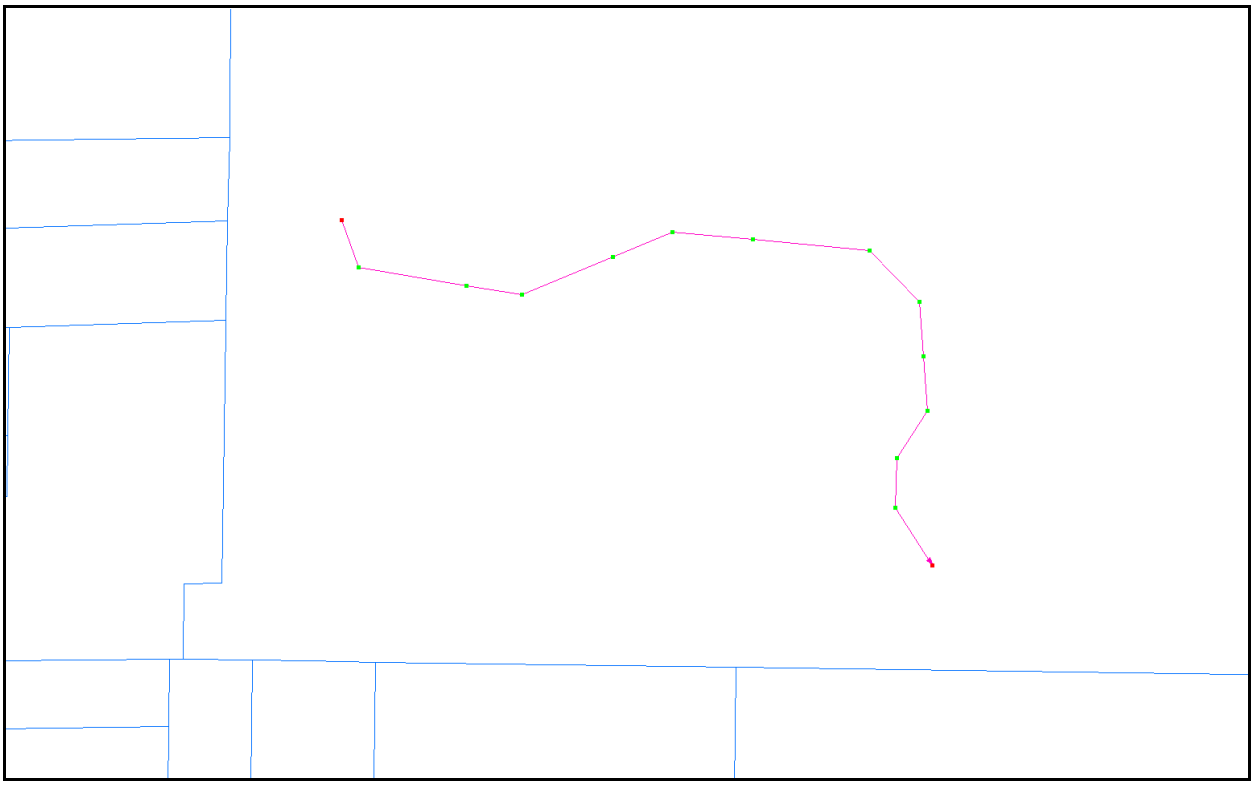


Figure 87 - A fictitious street digitized using only tangents.

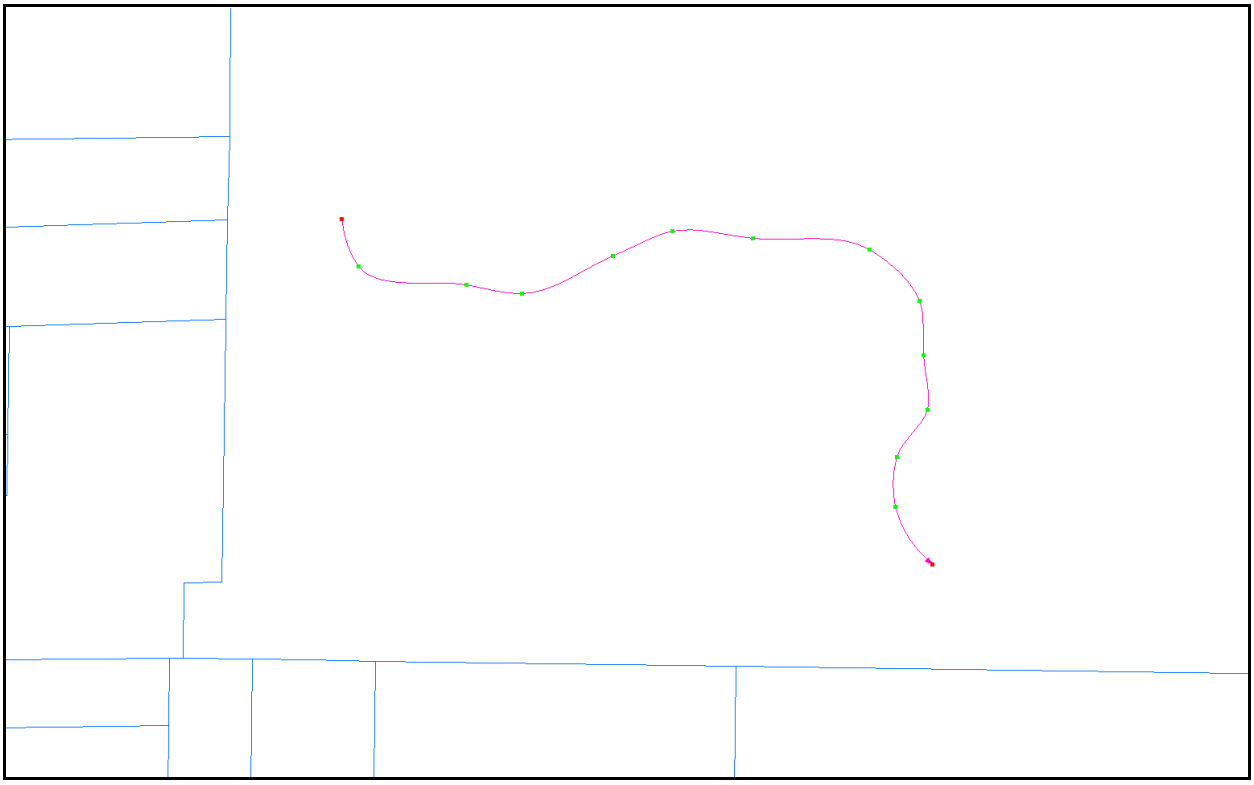


Figure 88 - The same street with a little smoothing applied.

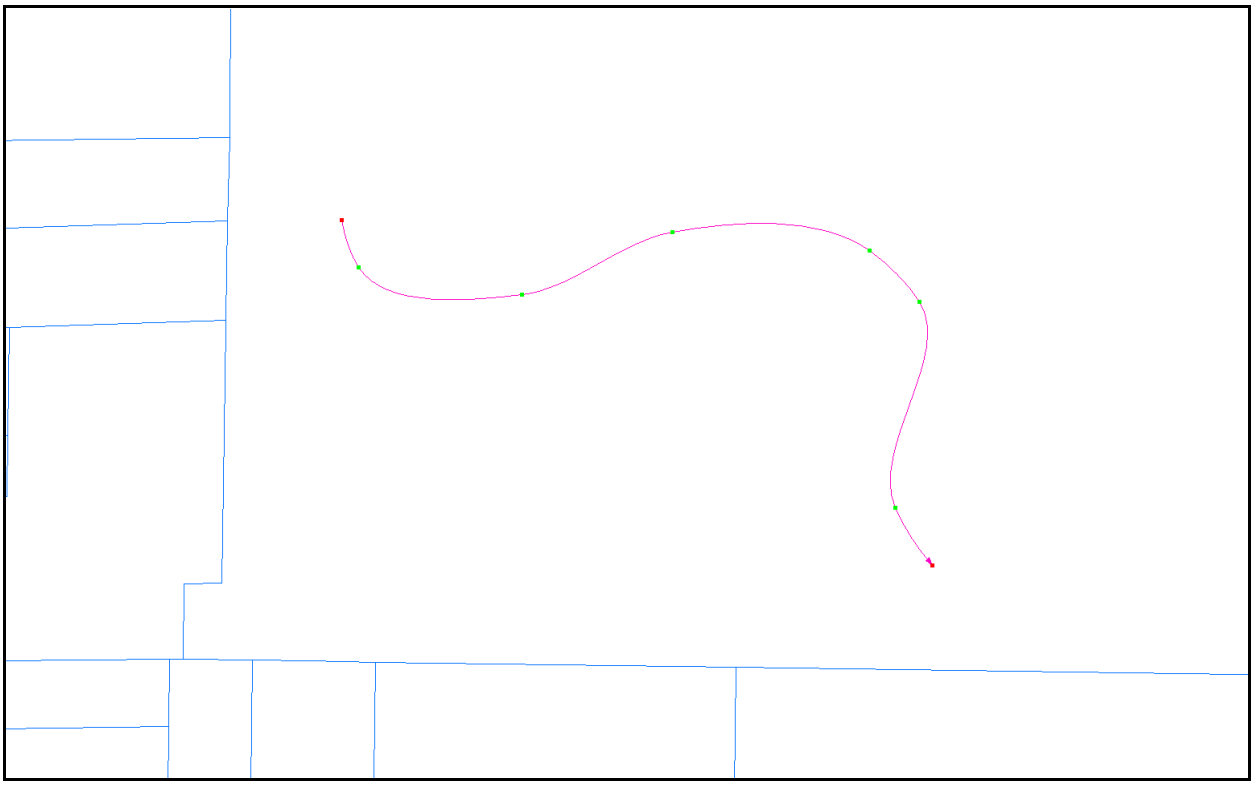
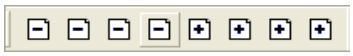


Figure 89 - The same street with more smoothing applied. Notice that the more smoothing is applied, the more vertices are deleted.

Tools I Created

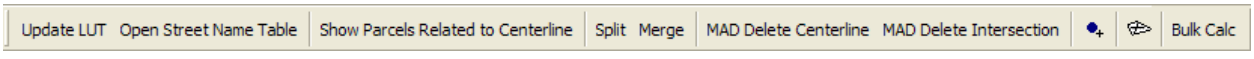
I created a custom zoom tool, so that I could zoom to predetermined magnifications. I used to have macros that did the same thing, and I wanted to retain their functionality. My custom zoom tool consists of eight buttons, and looks like this:



The first button zooms the display by a factor of -5, the next one by -4, the next one by -3, and the next one by -2. The buttons with the plus sign on them zoom by a factor of 2, 3, 4, and 5, respectively.

Tools Created by an Outside Consultant

We contracted a consultant to design some custom tools for us that would act as an interface between our GIS centerlines and our Master Address Database. (They also designed ones to work with our parcels and buildings, but I don't use those.) These custom tools are called simply the "Centerline MAD Tools," and are located on a toolbar that looks like this:



These tools were supplied in the form of an ArcMap extension entitled "MAD Tools," which I always have turned on. They are installed in my C:\Programs directory. The use of the tool buttons are explained in the chapter *Some Common Operations*, page 133.

Display Layers

You can see some of the layers in my Map Document in Figure 85. The table on the next several pages contains the complete list. The layers are listed in the table in the same order they appear in the Table of Contents (TOC). Of course, the list of layers is subject to change, but doesn't change a lot. The order for most of the layers in the TOC has been determined to be the most appropriate, based on my experience.

Some of the layer names are the original names from the database, and some of the names I have renamed in the TOC, for easier readability. I wasn't totally consistent in the way I renamed some of the layers, but their name is not that important, as long as I know what they are.

Of course, not all the layers are ever turned on at the same time, as it would slow the display to a crawl. Generally, only the layers that I need for whatever I'm working on at the moment will be turned on.

I employ a couple of conventions when it comes to labeling features. First, I always use the Tahoma font, as I find it pleasing. Also, for labeling unshaded polygons, I like to assign the same color to the font as I use for the polygon edge. This helps to mentally associate the labels with the proper polygons, when displaying polygons from multiple layers simultaneously.

Following the table is an explanation of how I use the individual layers.

Table of Contents Layers in my *Streets* Map Document (See Figure 85)

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
Streets	CCGIS.fcStreets	sdeDynamic database connection, CCGIS.IndyStreets feature dataset			Font Tahoma, Size 8, Color White, Field FULL_STNAME, don't display zoomed out beyond 1:15,000	Doesn't display when zoomed out beyond 1:120,000. Symbolized on the OPER_STATUS field. Below are listed the corresponding symbols used. All symbols contains arrowheads of matching color to indicate direction of digitization.
			White Line	1		OPER_STATUS = 'BUILT'
			Cyan Line	1		OPER_STATUS = 'PLANNED'
			Magenta Line	1		OPER_STATUS = 'PLATTED'
			Light Purple Line	1		OPER_STATUS = 'PLATTED/LOCATOR'
			Green Line	1		OPER_STATUS = 'REMOVED'
			Yellow Line	1		OPER_STATUS = 'VACATED'
			Orange Line	1		OPER_STATUS = 'VACATED/BUILT'
			Light Cyan Line	1		OPER_STATUS = 'VACATED/LOCATOR'
clxx_xxxxxx.dgn Polyline	clxx_xxxxxx.dgn Polyline	H:\GIS\NewCLs	Blue Line	1	N/A	Temporary layer. Batch of new centerlines from Brian Schneider. Any time a layer of new centerlines is to be added to the centerline layer, the layer will appear here in the TOC.
OutOfCountyStreets	CCGIS.fcStreetsOOC	sdeDynamic database connection, CCGIS.IndyStreets feature dataset			Font Tahoma, Size 8, Color White, Field FULL_STNAME, doesn't display zoomed out beyond 1:15,000	Uses same symbology as *Streets* layer, except built streets are black.
CCGIS.IndyStreets_ Topology	CCGIS.IndyStreets_ Topology	sdeDynamic database connection, CCGIS.IndyStreets feature dataset			N/A	Topology for centerlines and associated layers. See list below for symbology.
			Pink Line	5		Line errors. Single symbol.
			Red Square	8		Point errors. Single symbol.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
			Light Green Line	5		Line exceptions. Single symbol.
			Green Square	5		Point exceptions. Single symbol.
			Polygon with periwinkle cross-hatch shading	1		Dirty areas. Shaded polygons.
CCGIS.fcIntersection	CCGIS.fcIntersection	sdeDynamic database connection, CCGIS.IndyStreets feature dataset	N/A	N/A	N/A	Doesn't display when zoomed out beyond 1:20,000. See the table on page 73 for symbology.
CCGIS.fcCulDeSac	CCGIS.fcCulDeSac	sdeDynamic database connection, CCGIS.IndyStreets feature dataset			N/A	Doesn't display when zoomed out beyond 1:50,000. See list below for symbology.
			White circle with black halo of size 2	12	N/A	OPER_STATUS = 'BUILT'
			Magenta circle with black halo of size 2	12	N/A	OPER_STATUS = 'PLATTED'
			Yellow circle with black halo of size 2	12	N/A	OPER_STATUS = 'VACATED'
			Green circle with black halo of size 2	12	N/A	OPER_STATUS = 'PLANNED'
CCGIS.fcCulDeSacOOC	CCGIS.fcCulDeSacOOC	sdeDynamic database connection, CCGIS.IndyStreets feature dataset	White circle with black halo of size 2	12	N/A	OPER_STATUS = 'BUILT' Only built cul-de-sacs are mapped outside Marion County.
CCGIS.Parcels	CCGIS.fcParcels	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	Unshaded polygons with blue edges	1	Font Tahoma, Size 8, Color White, Field STNUMBER, don't display zoomed out beyond 1:12,000	Doesn't display when zoomed out beyond 1:30,000.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
CCGIS.Seclines	CCGIS.Seclines	sdeDynamic database connection, CCGIS.IndyStreets feature dataset	Green Line	6	N/A	Section lines (Public Land Survey System).
CCGIS.SurveySectionCorners	CCGIS.SurveySectionCorners	sdeDynamic database connection, CCGIS.IndyStreets feature dataset				Section corners (Public Land Survey System). See list below for symbology.
			Yellow Triangle	30	N/A	TYPE = 'CORNER'
			Green Triangle	30	N/A	TYPE = 'QUARTER-CORNER'
			Pink Triangle	30	N/A	TYPE = 'QUARTER-QUARTER-CORNER'
CCGIS.PreliminaryAlleys	CCGIS.PreliminaryAlleys	sdeDynamic database connection, CCGIS.IndyStreets feature dataset	Purple Line	5	N/A	Doesn't display when zoomed out beyond 1:50,000.
CCGIS.PreliminaryAlleysToo	CCGIS.PreliminaryAlleysToo	sdeDynamic database connection, CCGIS.IndyStreets feature dataset	Orange Line, 60% transparency	8	N/A	Doesn't display when zoomed out beyond 1:50,000.
CCGIS.Railroads	CCGIS.Railroad	sdeStatic database connection	White Line (railroad symbol)	4	N/A	Doesn't display when zoomed out beyond 1:50,000.
CCGIS.Pavement	CCGIS.Pavement	sdeStatic database connection	Green Line	1	N/A	Edges of paved roads. Doesn't display when zoomed out beyond 1:50,000.
CCGIS.MilePosts	CCGIS.MilePosts	sdeStatic database connection	Yellow Circle	10	Font Tahoma, Size 12, Color Yellow, Field MILE, doesn't display zoomed out beyond 1:50,000	Doesn't display when zoomed out beyond 1:50,000.
Zipcodes, Census Tracts, Townships (Group)						This is a Group Layer, consisting of the three layers listed below. The layers were combined because I frequently want to turn them all on or off simultaneously. Doesn't display when zoomed out beyond 1:50,000.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
CCGIS.ZipCodes	CCGIS.ZipCodes	sdeStatic database connection	Unshaded polygons outlined in green, 60% transparency	9	Font Tahoma, Size 20, Color Green, Field ZIPCODE	Production zipcode layer.
CCGIS. CensusTracts2000	CCGIS.CensusTracts2000	sdeStatic database connection	Unshaded polygons outlined in purple, 60% transparency	9	Font Tahoma, Size 20, Color Purple, Field TRACT	Census tracts, as they existed in the year 2000.
CCGIS. TownshipReference	CCGIS.TownshipReference	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	Unshaded polygons outlined in red, 60% transparency	12	Font Tahoma, Size 24, Color Red, Field TOWNSHIP	Marion County townships.
CCGIS.DmdBaseMap	CCGIS.DmdBaseMap	sdeStatic database connection	Unshaded polygons outlined in red	6	Font Pristina, Size 48, Color Red, Field BASEMAP	DMD Basemap boundaries.
Nine-County Zipcodes, Townships, Counties (Group)						This is a Group Layer, consisting of the three layers listed below. The layers were combined because I frequently want to turn them all on or off simultaneously. Doesn't display when zoomed out beyond 1:50,000.
Nine-County Zipcodes	Zipcodes.shp	D:\GIS\Layers\Shapes\ Custom\Nine County	Unshaded polygons outlined in green	5	Font Courier New Bold, Size 20, Color Green, Field ZIP	Zipcode boundaries in the "Nine County" area. This area consists of Marion County and its surrounding counties (Boone, Hamilton, Madison, Hendricks, Hancock, Morgan, Johnson, and Shelby Counties). The boundaries in Marion County are not as accurate as in the CCGIS.ZipCodes layer.
Nine-County Townships	Townships.shp	D:\GIS\Layers\Shapes\ Custom\Nine County	Unshaded polygons outlined in red	5	Font Tahoma, Size 20, Color Red, Field NAME	Township boundaries in the "Nine County" area. The boundaries in Marion County are not as accurate as in the CCGIS.TownshipReference layer.
Nine-County Counties	Boundary.shp	D:\GIS\Layers\Shapes\ Custom\Nine County	Unshaded polygons outlined in orange	12	Font Tahoma, Size 20, Color Orange, Field NAME	County boundaries in the "Nine County" area. The boundaries in Marion County are not as accurate as in the CCGIS.TownshipReference layer.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
CCGIS. fcPlattedSubdivision	CCGIS.Subdivisions (Platted)	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	Unshaded polygons outlined in yellow	6	Font Tahoma, Size 12, Color Yellow, Field DEVELOPMENT_NAME, doesn't display zoomed out beyond 1:15,000	Subdivision boundaries. Shows boundaries of each section. Doesn't display when zoomed out beyond 1:50,000.
CCGIS. fcCommonSubdivision	CCGIS.Subdivisons (Common)	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	Unshaded polygons outlined in yellow	6	Font Tahoma, Size 12, Color Yellow, Field DEVELOPMENT_NAME, doesn't display zoomed out beyond 1:15,000	Subdivision boundaries. Shows boundaries of entire subdivisions without their individual sections. Doesn't display when zoomed out beyond 1:50,000.
CCGIS.Parks	CCGIS.Parks	sdeStatic database connection	Polygons shaded in olive green	N/A	Font Tahoma, Size 10, Color Dark Olive Green, Field NAME	
2007 AERIALS	CCGIS.Image2007	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 2007, rectified and mosaiced. At the time of this writing, these were the most recent available. The layer name in the TOC is capitalized to make it stand out. This is because there are many aerial photo layers in the Map Document, and I want to be able to quickly locate the most recent.
Historical Aerial Photos (Group)						This is a Group Layer, consisting of the 20 layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the TOC display, since I only turn one of them on at a time. All the photos in these layers were rectified to line up with our data. They are listed in chronological order.
2006 Aerials	[Multiple]	A total of 161 Individual files in MrSID format. Each file is one photo. Some are stored on my C: drive, some on my D: drive, and some on my H: (a network) drive.	N/A	N/A	N/A	Aerial photos (color) taken in 2006. These were supplied by Pictometry. They were taken at the same time they took the oblique photography. They were not mosaiced. I stored as many of the files on my harddrives as I could fit (for faster display time), and stored the rest on a network server. (The average file size is about 400 MB.)
CCGIS.Image2005	CCGIS.Image2005	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 2005. Mosaiced.
CCGIS.Image2004	CCGIS.Image2004	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 2004. Mosaiced.
CCGIS.Image2003	CCGIS.Image2003	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 2003. Mosaiced.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
CCGIS.Image2002	CCGIS.Image2002	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 2002. Mosaiced.
CCGIS.Image2001	CCGIS.Image2001	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 2001. Mosaiced.
CCGIS.Image2000	CCGIS.Image2000	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 2000. Mosaiced.
CCGIS.Image1999	CCGIS.Image1999	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1999. Mosaiced.
CCGIS.Image1997	CCGIS.Image1997	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1997. Mosaiced.
CCGIS.Image1995	CCGIS.Image1995	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1995. Mosaiced.
CCGIS.Image1993	CCGIS.Image1993	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1993. Mosaiced.
CCGIS.Image1986	CCGIS.Image1986	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (color) taken in 1986. Mosaiced.
CCGIS.Image1979	CCGIS.Image1979	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1979. Mosaiced.
CCGIS.Image1978	CCGIS.Image1978	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1978. Mosaiced.
CCGIS.Image1972	CCGIS.Image1972	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1972. Mosaiced.
County_1971.tif	County_1971.tif	D:\GIS\Layers\Images\ Aerials	N/A	N/A	N/A	A poster-sized, scanned, countywide aerial photo (color), taken by Indianapolis Power & Light Company.
CCGIS.Image1962	CCGIS.Image1962	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1962. Mosaiced.
CCGIS.Image1956	CCGIS.Image1956	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1956. Mosaiced.
CCGIS.Image1941	CCGIS.Image1941	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1941. Mosaiced.
CCGIS.Image1937	CCGIS.Image1937	sdeStatic database connection	N/A	N/A	N/A	Aerial photos (black and white) taken in 1937. Mosaiced.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
Out-of-County Centerlines (Group)						This is a Group Layer, consisting of the eight layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the TOC display, since I only turn one of them on at a time.
Boone	inboonst_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inboon\BooneCoSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Boone Co. provided by TeleAtlas in April, 2007.
Hamilton	inhamist_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inhami\HamiltonSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Hamilton Co. provided by TeleAtlas in April, 2007.
Hancock	inhamist_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inhanc\HancockCoSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Hancock Co. provided by TeleAtlas in April, 2007.
Hendricks	inhendst_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inhend\HendricksSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Hendricks Co. provided by TeleAtlas in April, 2007.
Johnson	injohnst_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ injohn\JohnsonSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Johnson Co. provided by TeleAtlas in April, 2007.
Madison	inmadist_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inmadi\MadisonSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Madison Co. provided by TeleAtlas in April, 2007.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
Morgan	inmorg_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inmorg\MorganSPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Morgan Co. provided by TeleAtlas in April, 2007.
Shelby	inshel_SPE	M:\Shared\IMAGIS\ TeleAtlas0407\usa\in\ inshel\Shelby_SPE\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	Centerlines for Shelby Co. provided by TeleAtlas in April, 2007.
Historical Maps (Group)						This is a Group Layer, consisting of the 13 layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the TOC display, since I only turn one of them on at a time. These layers were all rectified to line up with our data. They are listed in chronological order.
1968 City.jpg	City1968.jpg	D:\GIS\Layers\Images\ Maps	N/A	N/A	N/A	Scanned and rectified 1968 map (color) of Indianapolis. The map belongs to Brian Schneider.
1959 City.jpg	City1959.jpg	D:\GIS\Layers\Images\ Maps	N/A	N/A	N/A	Scanned and rectified 1959 map (color) of Indianapolis. The map belongs to Brian Schneider.
1911 Inner City.sid	1911.sid	D:\GIS\Layers\Images\ Maps\1911\	N/A	N/A	N/A	At the time I last revised this document, I could not get this layer to display, and I don't remember the details on this layer.
1909 Center Twp.sid	Baist1909Cent.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Center Township, in MrSID format.
1909 Decatur Twp.sid	Baist1909Deca.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Decatur Township, in MrSID format.
1909 Franklin Twp.sid	Baist1909Fran.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Franklin Township, in MrSID format.
1909 Lawrence Twp.sid	Baist1909Lawr.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Lawrence Township, in MrSID format.
1909 Perry Twp.sid	Baist1909Perr.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Perry Township, in MrSID format.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
1909 Pike Twp.sid	Baist1909Pike.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Pike Township, in MrSID format.
1909 Warren Twp.sid	Baist1909Warr.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Warren Township, in MrSID format.
1909 Washington Twp.sid	Baist1909Wash.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Washington Township, in MrSID format.
1909 Wayne Twp.sid	Baist1909Wayn.sid	M:\Data\Images\IHPC\ Baist1909\	N/A	N/A	N/A	Scanned 1909 map from the G. W. Baist Company of Wayne Township, in MrSID format.
1898 City.jpg	1898 Rectified.jpg	D:\GIS\Layers\Images\ Maps\1898\	N/A	N/A	N/A	This layer is a scan of a framed map that Dave Surina used to have hanging on the wall in his office.
Historical Centerlines (Group)						This is a Group Layer, consisting of the three layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the TOC display, since I only turn one of them on at a time.
Streets20060126	Streets20060126.shp	H:\GIS\Layers\Shapes\ Custom\County\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	This is a snapshot (copy) of the centerlines, as they existed on 1/26/2006.
cl030908	cl030908.shp	H:\GIS\Layers\Shapes\ Custom\County\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	This is a snapshot (copy) of the centerlines, as they existed on 9/8/2003.
cl1997	cl1997.shp	H:\GIS\Layers\Shapes\ Custom\County\	Black Line	1	Font Tahoma, Size 8, Color Black, Expression [NAME] + " " + [TYPE] + " " + [SUFFIX], don't display zoomed out beyond 1:15,000	This is a snapshot (copy) of the centerlines, as they existed sometime in 1997.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
Plat Indexes (Group)						This is a Group Layer, consisting of the seven layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the display, since I only turn one of them on at a time.
29Cent_index	Cent1929_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light yellow-green	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1929 maps of portions of Center Township. This layer consists of polygons showing the extents covered by the individual maps.
56Dec_index	Deca1956_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in blue-gray	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1956 maps of portions of Decatur Township. This layer consists of polygons showing the extents covered by the individual maps.
43Dec_index	Deca1943_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in aquamarine	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1943 maps of portions of Decatur Township. This layer consists of polygons showing the extents covered by the individual maps.
22Dec_index	Deca1922_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in blue-gray	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1922 maps of portions of Decatur Township. This layer consists of polygons showing the extents covered by the individual maps.
22Per_index	Perr1922_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in olive green	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1922 maps of portions of Perry Township. This layer consists of polygons showing the extents covered by the individual maps.
39WashPart1_index	Wash1939Part1_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in lavender	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1939 maps of portions of Washington Township. This layer consists of polygons showing the extents covered by the individual maps.
39WashPart2_index	Wash1939Part1_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light blue	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1939 maps of portions of Washington Township. This layer consists of polygons showing the extents covered by the individual maps.
42Way_index	Wayn1942_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in aquamarine	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1942 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
31WayPart1_index	Wayn1931Part1_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light green	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1931 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.
31WayPart2_index	Wayn1931Part2_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in turquoise	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1931 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.
31WayPart3_index	Wayn1931Part3_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in sea green	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1931 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.
31WayPart4_index	Wayn1931Part4_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light blue	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1931 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.
1891Way_index	Wayn1891_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light yellow	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned 1891 maps of portions of Wayne Township. This layer consists of polygons showing the extents covered by the individual maps.
Book1A_index	Book1A_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in periwinkle	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Book2_index	Book2_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in turquoise	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Book3_index	Book3_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in light blue	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Book4_index	Book4_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in pink	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
Book4A_index	Book4A_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in periwinkle	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Book5_index	Book5_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in sea green	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Book5A_index	Book5A_index.shp	D:\GIS\Layers\Plat Indexes	Polygon shaded in light blue	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is a polygon showing the extent of a scanned 1956 index map of Indianapolis produced by the Sanborn Company. The index map references individual maps which are contained in the Sanborn_index.shp layer described below.
Sanborn_index	Sanborn_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in lavender	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	This layer is an index layer for the individual scanned maps from the Sanborn Company referenced in the above seven layers. It consists of polygons showing the extents covered by the individual maps.
Address_index	Address_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in light blue	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned address maps of Marion County, vintage approximately 1956. This layer consists of polygons showing the extents covered by the individual maps.
Swrbk_index	Swrbk_index.shp	D:\GIS\Layers\Plat Indexes	Polygons shaded in aquamarine	N/A	Font Tahoma, Size 8, Color Black, Field IMAGENAME	Index layer for some scanned Sewer Book maps of Marion County. This layer consists of polygons showing the extents covered by the individual maps.
Center 1929 Plats (Group)						This is a Group Layer, containing about 40 scanned and rectified 1929 plats of Center Township, in MrSID format. The layers were combined because putting them together in a Group Layer is an easy way to condense the display, since I only turn one of them on at a time. I have only listed one of the layers below, as an example. All the plats are located in the same directory.
Cent29sht12.sid	cent29sht12.sid	D:\GIS\Layers\Images\Maps\Assessors\Center 1929\	N/A	N/A	N/A	Scanned and rectified 1929 map (black and white) of a portion of Center Township, in MrSID format.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
USGSQUAD	usgsquad.dbf	M:\DATA\Images\Maps\	N/A	N/A	N/A	Image catalog of scanned and rectified USGS 1:24,000-scale topographic maps covering Marion County.
Former Towns	Former Towns.shp	D:\GIS\Layers\Shapes\ Custom\County\	Black Star Outline	18	Font Tahoma, Size 12, Color White, Field NAME	Point layer showing locations of former towns in Marion County. These are towns that got swallowed up by "Unigov," and that are no longer towns.
Airports, Towns, Excluded Cities (Group)						This is a Group Layer, consisting of the three layers listed below. The layers were combined because I frequently want to turn them all on or off simultaneously.
CCGIS.Airports	CCGIS.Airports	sdeStatic database connection	Polygons shaded in dark gray	N/A	Font Tahoma, Size 14, Color White, Field NAME	Parcel boundaries of airports.
CCGIS.Towns	CCGIS.Towns	sdeStatic database connection	Polygons shaded in dark gray	N/A	Font Tahoma, Size 14, Color White, Field CITYNAME	Parcel boundaries of the remaining towns within Marion County. Although they fall within the boundary of "Unigov," they still retain their own governing bodies.
CCGIS.Exclcity	CCGIS.Exclcity	sdeStatic database connection	Polygons shaded in dark gray	N/A	Font Tahoma, Size 14, Color White, Field CITYNAME	Parcel boundaries of the four excluded cities in Marion County (see page 39).
Misc. MAD Layers (Group)						This is a Group Layer, consisting of the three layers listed below. The layers were combined because putting them together in a Group Layer is an easy way to condense the display. I never turn these layers on, as I don't use them, but they are required to be present, in order for the centerline MAD Tools to work properly.
Cities	Cities.shp	D:\GIS\Layers\Shapes\ Custom\Nine County	N/A	N/A	N/A	This is a custom layer, created specifically for use with the MAD Tools. It basically contains the boundary of the nine-county area, as well as the boundaries of the excluded cities within that area.
CCGIS.fcBuildings	CCGIS.fcBuildings	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	N/A	N/A	N/A	Buildings. These are abstract polygons that have verified addresses attached to them. Eventually, this layer will be combined with our other *Buildings* layer which contains accurate polygons.

TOC Name	Layer Name	Source	Symbol Type	Symbol Size	Labels	Remarks
CCGIS.Buildings	CCGIS.Buildings	sdeStatic database connection	Polygons shaded in light gray with dark gray edges	1	N/A	Building footprints. Doesn't display when zoomed out beyond 1:25,000.
CCGIS.CensusBlocks 2000	CCGIS.CensusBlocks2000	sdeStatic database connection	N/A	N/A	N/A	Census tract boundaries, as they existed in the year 2000.
CCGIS.fcUnits	CCGIS.fcUnits	sdeDynamic database connection, CCGIS.IndyParcels feature dataset	N/A	N/A	N/A	Point layer showing individual building units (e.g., apartments, condominiums, and duplexes). The unit addresses are included as attributes.

Streets - At the top of the list in the TOC is the *Streets* layer, the main layer that I edit. I like to keep this layer at the top, since it is the layer that I edit most, and also because when I open the ESRI dialog box to search by attributes, the layer at the top of the Table of Contents (TOC) is listed first in the dropdown list of layers that you can select from. This reduces the number of steps to do a search, since I most often search on the *Streets* layer.

Centerlines are usually labeled with their full street names. I almost never turn off the labels while editing.

clxx_xxxxxx.dgn Polyline - A lot of times the TOC will contain one or more layers of new centerlines from Brian Schneider, just below the *Streets* layer. (This is one example.) These temporary layers are stored in my H:\GIS\NewCLs folder, as I said before. Besides the Polyline layer, which contains the geometry, there is always an accompanying Annotation layer in the .dgn files for each set of centerlines. Brian uses the Annotation layers to provide me the street names and address ranges for the new centerlines. The Annotation layer is not present in the TOC here because the street names and address ranges have already been transferred onto the new centerlines, and so the Annotation layer is no longer needed and was deleted.

The reason this Polyline layer is still present in my TOC is that the process of adding new centerlines from Brian is a multi-step process, which I will explain.

First, when Brian notifies me by e-mail he has some new centerlines for me, I paste them into the *Streets* layer, clean up the geometry, topology, and attributes, and then notify him by reply e-mail that those streets are in. Brian will then add the parcels for those streets, with their addresses and other attributes, into the *Parcels* layer. (Recall from the discussion on page 3 that Brian can't add the parcels until the centerlines are in, because the way we have our Master Address Database set up, a parcel address can't be entered unless there is a valid centerline containing that address already in the system.)

There is typically a day or two between the time I first put in the centerlines, till the time Brian finishes the parcels. A few days after putting in the new centerlines, I will return to the area to see if Brian has finished putting in the parcels. (To see the changes, I have to reconcile my version with the DEFAULT version.) If he has, I review all the address ranges on the centerlines, to see if any of them should be adjusted, in order to more closely match the parcel addresses. At that time I also populate the COORDINATE and COORDDIR attributes, because it's easier to determine the values for these attributes after the adjacent parcels with their addresses are visible. Until the time that I return to the new centerlines to put the finishing touches on the address ranges and coordinate attributes, I keep the layers for the new centerlines Brian sent me in my TOC, both to remind me that they're not finished yet, and also so I can zoom to their extent, to quickly locate them again. Once the changes to the new centerlines are complete, I delete those layers from my Map Document.

These kinds of temporary layers are only turned on when editing the new centerlines supplied by them.

OutOfCountyStreets - The *StreetsOOC* layer. This layer is next because it is almost as important as the *Streets* layer; in fact, it is basically the same layer, just with a different geographic extent. Normally, I leave this layer turned off, unless I need to edit it, or I'm editing centerlines in the *Streets* layer very close to the edge of the county. This layer is labeled using the FULL_STNAME field, just like the *Streets* layer.

CCGIS.IndyStreets_Topology, **CCGIS.Intersection**, **CCGIS.CulDeSac**, **CCGIS.fcCuldeSacOOC** - These are the main layers that I edit in conjunction with the centerlines, so they need to be near the top of the list. The *Topology* layer is usually not turned on until I'm ready to validate the topology. The *Intersection* and *CuldeSacs* layers may or may not be turned on, depending on what I'm working on. These layers are not labeled.

Parcels - This is the master parcel layer, the one that Brian edits, and it is on the *SDEDynamic* server. I have this layer turned on almost all the time while editing centerlines. Without a doubt, it is the most important and most-often-used background layer I have at my disposal. Probably 90% of all the centerlines are based on this layer. The parcels are labeled with their address, because that is what the centerline address ranges rely on (except for private streets).

CCGIS.Seclines, **CCGIS.SurveySectionCorners** - These two layers (Section Lines and Section Corners), are two of the fundamental layers that the topology for the centerlines and related layers is built on. Because of this, it is sometimes helpful to be able to display these layers underneath the *Topology* layer. Most of the time they are not turned on, and they are not labeled.

CCGIS.PreliminaryAlleys, **CCGIS.PreliminaryAlleysToo** - See page 88.

CCGIS.Railroads - Used to display the railroads. Seldom used. Not labeled.

CCGIS.Pavement - Occasionally, I use this layer to help spot certain pavement configurations, such as roundabouts, streets with medians, or apartment streets. For these types of items, for which I have no attribute to search on and I have to do a visual search, it is faster to use this layer than the aerial photos. No labels.

CCGIS.Mileposts - This layer is used when assigning milepost numbers to the interstate centerlines. Turned off otherwise. Labeled using the milepost numbers.

Zipcodes, Census Tracts, Townships (Group) - This Group Layer serves as a reminder that the centerline attributes change if the centerlines cross these polygon boundaries. Zipcode and census tract polygons are labeled with their respective numbers, and township polygons are labeled with their names. The labels are used to help populate the appropriate centerline attributes. Townships are displayed in a thicker line than the other two layers, and displayed underneath them. All three boundaries are displayed with 60% transparency. In this way, if the polygon boundaries of the three layers overlap, all lines will still be visible. I normally only turn on this layer when populating centerline attributes.

DmdBaseMap - This layer is normally used when I'm entering Street Acceptances or Vacations. Since I file these documents by DMD Basemap, I need to look up the basemap number of the area where the affected centerlines are located, in order to write that number on the first page of the document. Polygons are labeled with the basemap number.

Nine County Zipcodes, Townships, Counties (Group) - This Group Layer was used when I initially populated the centerlines outside Marion County. There are still some centerlines outside the county for which these values are not populated, so when I edit them, I depend on these layers. I also use this layer every time I update the centerlines when we receive a new set of aerial photos. When this happens, I turn on the aerials with only the nine-county boundaries and centerlines displayed on top, and then sweep back and forth across the county and look for changes. The county boundaries let me know I've reached the end of the county and I need to pan my display to the next extent.

The county boundaries are drawn first, with the township boundaries above them, and the zipcodes above the townships. The township boundaries are drawn with a wider line than the zipcodes, and the county boundaries are drawn with an even wider line. In this way, if the polygon boundaries of the three layers overlap, all lines will still be visible. The zipcode polygons are labeled with their zipcode numbers, and the townships and counties are labeled with their names.

CCGIS.PlattedSubdivision - I sometimes use this layer to verify why centerlines in subdivisions are split. Normally Brian gives me new centerlines in subdivisions in (subdivision) sections, because that is normally how they are approved and recorded. This means that a centerline for a certain street could be split for no apparent reason, other than the fact that the two parts of the street belong to different sections of the subdivision. Frequently, this means that each section will be accepted by the City on a different date, which means the centerlines will need to remain split, because the acceptance dates will be different. In the case where adjacent sections of the same subdivision are accepted on the same date, the two adjacent centerlines of the same street in different sections can and will be merged. The layer is labeled with the names of the subdivisions, including their sections.

CCGIS.fcCommonSubdivision - This layer is used in the few cases where I need to locate a subdivision, and I don't care about the individual sections. Labeled with the subdivision names.

CCGIS.Parks - Since streets within parks are the county's responsibility, I need to see the

boundaries of the parks in order to populate the OPER_STATUS attribute for these streets correctly.

2007 AERIALS - This spot in the TOC is reserved for the most recent layer of aerial photos, which at the time of this writing was the photos taken in 2007. As new photos are acquired, they are placed in this spot, and the old layer is moved to the Group Layer listed below. This makes it easy to always find the most recent set of aerials in the TOC.

As you can imagine, this layer is used quite a lot. Two of its main uses are verifying whether or not streets (and alleys) are improved, and digitizing centerlines for private streets for which there is no right-of-way. The layer is not turned on when not needed.

Historical Aerial Photos (Group) - The individual layers in this Group Layer are used to digitize centerlines of historical streets for which neither the pavement nor right-of-way exists. They are also sometimes used to clear up questions as to the history of changes to individual centerlines.

Out-of-County Centerlines (Group) - These layers are used when working with centerlines outside Marion County. I may use them to copy and paste, or just to verify attributes (such as street names and address ranges).

Historical Maps (Group) - These are mostly used to help verify names of historical streets, but sometimes useful for trying to figure out the correct name of existing streets, since older maps tend to be more accurate than current ones. (This is because errors can be introduced over time (human error), and because street names sometimes change through common usage, due to ignorance or laziness.)

Historical Centerlines (Group) - These are sometimes used to clear up various questions about the centerlines. Another use is when I accidentally delete a centerline and save my changes before realizing I need the centerline back (including the same Tag number). They may also be used to replace missing centerlines.

By missing centerlines I am talking about centerlines that used to exist in the database, but don't now. Unfortunately, it has been my experience that about once or twice a year, I will come across a place in the database where a centerline should exist, but it doesn't, and I know that the centerline *used to exist*. Apparently, the database gets slightly corrupted from time to time, for whatever reason. The centerline needs to be replaced and, If possible, retain its original Tag number.

Plat Indexes (Group) - This Group Layer contains indexes to various sets of scanned historical maps. Just like the historical maps listed above, I use these historical plats to help verify street names and alignments. These indexes contains polygons showing the extents of the various maps, which frequently overlap or contain missing areas (they do not form a mosaic). Labeled with the names of the individual map files which cover a certain area.

Center 1929 Plats (Group) - These are more historical plats that are helpful in verifying old street names and alignments.

UGSSQUAD - Seldom used, but useful to verify the general location of PLSS section lines.

Former Towns - Of no use for editing centerlines. This layer is in my map document only because I created it, and I like to look at it from time to time!

Airports, Towns, Excluded Cities (Group) - This Group Layer serves as a reminder that the centerline attributes change if the centerlines cross these polygon boundaries. The polygons of all three layers are labeled with their respective names. The labels are also used to help populate the appropriate centerline attributes. This Group Layer is nearly always left turned on, partly because it contains few features and it displays quickly.

Misc. MAD Layers (Group) - Most of these layers are in the TOC only because they are required for proper operation of the Centerline MAD Tools. They are never used for display; that is why they are at the very bottom of the TOC.

The *Cities* and *CCGIS.CensusBlocks2000* layers are used for automatic populating of the CITY_LEFT

and CITY_RIGHT and L_TRACT and R_TRACT attributes. (See the section entitled "Automatic Populating of Attributes" on page 136.) The *CCGIS.fcBuildings* and *CCGIS.fcUnits* layers are used by the MAD parcel-editing and building-editing tools, which Brian Schneider uses but I don't. However, the *CCGIS.fcUnits* layer is very useful when I am populating address ranges in apartment and condominium complexes, because it allows me to see all the individual unit addresses.

I use the *CCGIS.Buildings* layer whenever I want to see the building footprints, which is seldom. Although this layer doesn't participate in the MAD tools, I include it here because the *CCGIS.fcBuildings* layer is also here.

The Right Monitor Display

As you can see from Figure 86, all the available monitor space is taken up by four windows. The one on the left is the generic ESRI attribute inspector. In the middle is the Identify Results window, and on the right is the Snapping window. Most of the time the only layers that I snap to are the first three layers in the Snapping window, but I frequently change which of the three types of snapping (vertex, edge, node) I use for each of these three layers. That is why I like to keep this window visible at all times.

If you look closely you will see a portion of another window behind the first and second windows. This is the *CMFI* window I've referred to earlier. This is actually a fourth window I like to keep open on this screen. Depending on what attributes I'm editing, I will either use ESRI's window or the *CMFI*, so I like to have access to both of them at all times. (More on that in the next chapter.) If the *CMFI* is not frontmost, as in Figure 86, I simply click the thin sliver of it that's visible to bring it to the front. Notice that the ESRI Attribute window extends farther down than the *CMFI*. This is so when I'm ready to bring it back to the front, the bottom of it will be visible behind the *CMFI*, so I can click on it.

Eliminating "Invisible" Centerlines

I have discovered in the course of my editing with ArcMap that, over time, ArcMap creates "invisible" centerlines — invisible centerlines that have attributes, but a zero length, and no geometry. How or why these get created I don't know, but they always show up after editing for several weeks. I find them by searching for the centerlines that have no Tag number assigned, using a pre-saved expression called "No_tag" (see next section.) Once I find them, of course, I delete them.

Expression Files

I have several SQL conditions that I use frequently when selecting centerlines in the Select by Attributes dialog. I have these conditions saved as ArcMap expression files in my D:\GIS\Centerlines\Geodb directory. They are:

Coming_off_bond.exp - Used to find those streets for which the OPER_STATUS value needs to be changed from "DEVELOPER" to "DPW." (See page 46.) The expression is:

DATE_ACCEPTED < SYSDATE - (365*3) AND MAINT_JURIS = 'DEVELOPER'

Extract _Cntrlin_layer.exp - Used to extract the *Cntrlin* layer from the *Streets* layer, in case I want to create a shapefile of the *Cntrlin* layer for a customer. The expression is:

OPER_STATUS = 'BUILT' OR OPER_STATUS = 'VACATED/BUILT' OR (OPER_STATUS = 'PLATTED' AND MAINT_JURIS = 'DEVELOPER') OR (OPER_STATUS = 'PLATTED' AND MAINT_JURIS = 'DEVELOPER')

No_full_stname.exp - Used to find those streets for which the FULL_STNAME field is blank. The expression is:

FULL_STNAME IS NULL OR FULL_STNAME = ''

No_oper_status.exp - Used to find those streets for which the OPER_STATUS field is blank. The expression is:

OPER_STATUS IS NULL OR OPER_STATUS = ''

No_spd_lim.exp - Used to find those streets for which the SPD_LIM field is blank. The expression is:

OPER_STATUS = 'BUILT' AND (COUNTY_LEFT = 'MARION' OR COUNTY_RIGHT = 'MARION') AND (SPD_LIM = 0 OR SPD_LIM IS NULL)

No_street_name.exp - Used to find those streets for which the STREET_NAME field is blank. The expression is:

STREET_NAME IS NULL OR STREET_NAME = ''

No_tag.exp - Used to find those streets for which the CENTERLINE_TAG field is blank, including "invisible" centerlines. This expression is:

CENTERLINE_TAG IS NULL OR CENTERLINE_TAG = 0

Vacations_in_remarks.exp - Used to find those streets for which the REMARKS field contains the value "VAC." I used to put supporting documentation for street vacations in the REMARKS field, before the VACATED field was created. A typical vacation petition number would be like "89-VAC-21," indicating the 21st vacation in the year 1989. By searching on the REMARKS values that contained "VAC," I found all these entries and moved the value over to the VACATED field. Therefore, this expression is no longer needed. It is:

REMARKS LIKE '%VAC%'

Layer Files

The most common way I like to symbolize the centerlines is by their OPER_STATUS field value. However, occasionally I like to symbolize them based on some other attribute, depending on what I'm doing. For this reason, I have created Layer Files to store various symbolization configurations based on these other attributes. These Layer Files are stored in my D:\GIS\Centerlines\Layer Files directory. A few of these are variations of my *Streets With Casings* Layer File. I do not want to go into a full discussion of the details of these files.

Here is the complete list. In the file names, "CL" stands for "centerline."

CL_aerials.lyr
CL_block_ID.lyr
CL_level.lyr

CL_Major Streets Level 0 Fill.lyr
CL_no_address.lyr
CL_no_tags.lyr
CL_one_way_dir.lyr
CL_rectifying.lyr
CL_spd_lim.lyr
CL_status.lyr
CL_statusOOC.lyr
CL_subtype.lyr
CL_weight_lim.lyr
Intersex_proc.lyr
Intersex_type.lyr
Route_Oldcounts.lyr
Route_Stclose.lyr
Route_Tcounts.lyr
Streets (SDE Dynamic) with casings.lyr
Streets (SDE Static) with casings (truncated).lyr
Streets (SDE Static) with casings.lyr
StreetsOOC (SDE Static) with casings.lyr

Fail-Safe Procedures

In the last year, more than ever, I have had occasional trouble with my *Streets* May Document not opening when I want to begin an editing session. When this happens, this is the process: I will load the generic ArcMap (the part that loads before you select a Map Document to load), and it loads fine. Then when the list of previously-used Map Documents appears, I select my *Streets* document to load. Usually, nothing will happen for a few minutes after I click the button to load the file (which is normal — my *Streets* document takes seven minutes to load!), but at some point the Windows message pops up that says this particular program has encountered a problem and needs to close (we're so sorry!). I have no idea what causes this.

In the past, when my Map Document would not load, I recreated it from scratch. I wanted to be able to recreate it as closely as possible to the way it existed before, so I hit on the idea of periodically capturing a screenshot of my loaded document. Then I could refer to the screenshot to know how to reconstruct my Document. I save these files as JPEG files in my D:\GIS\Centerlines directory, and I include the date of the screenshot in the filename (e.g., *Screenshot 20070914.jpg).* Periodically, I take a new screenshot of my Map Document, to keep up with the changes I make.

Still, the process of recreating my Document from scratch would take a very long time, since I have many layers in it, and also because I have complex symbology created for some of the layers. So then I hit upon the idea of saving each of the layers in my Document as a layer file. Then, all I would have to do to recreate my Document is to start with a blank Document, and then add each of the layer files in succession. By numbering the layer files in such a way as to know in what order they should be loaded, I could quickly reproduce my Map Document.

I periodically repeat this process of extracting my layers as Layer Files. Each time I do this, I put the Layer Files in a new directory under my D:\GIS\Centerlines\Layer Files directory. For example, the last time I did this, I put the files in a directory called *Map Layers 20071101*. Each layer file is given the default name, which is the same as the name of the layer in the ArcMap Table of Contents. In front of the default name I add a two-digit number, which denotes the order in the lineup that the layer goes in. For example, Section Lines are currently the eighth layer from the top in the Table of Contents, so the name of their Layer File is *08CCGIS.Seclines.lyr*. This naming convention not only tells me what order to place the Layer Files in, but also means the Layer Files will be sorted in the proper order in Windows Explorer, which makes a nice visual reference.

Finally, I have decided to keep periodic backups of my Map Document file, and include the date of the backup in the filename (e.g., *Streets 20071101*). I need to remember to keep up with this, because I am constantly making little tweaks to my *Streets* document.

Error Reporting

Recognizing that the centerlines will never be perfect, I accept reports of errors from anyone. I accept them by any means that people choose to give them to me, including verbally, by e-mail, phone calls, and through our agency's website. In the early days of maintaining centerlines, I used to receive a lot of error reports, mostly from staff. Now I hardly get them anymore, but I still take them seriously.

I always ask for supporting documentation before I will make a change to the centerlines. I usually don't simply take someone's word that something is wrong, but I want some kind of proof. If the error involves a street name, I consult the historical maps and other references I have access to to help verify the person's claim.

Filing System

I have a five-drawer filing cabinet that I use to store certain centerline-related documents in. These are all located in the fourth drawer down. There are four major divisions of documents, each separated by a green, heavy cardboard, tabbed divider. They are:

Error Reports
Acceptances
Vacations
Misc.

Documents in the first three categories are filed by DMD Basemap number. Since there are 51 basemaps, this means there are 51 folders in each division, one for each basemap. The documents are separated by basemap because there are so many of them. Whenever I process an error report, Street Acceptance or Vacation, on the top right corner of the first page of the document, I write the DMD Basemap number where the area described in the document is located. If an area crosses multiple basemaps, I list the one that contains the majority of the area involved.

Underneath the basemap number, I write the date I finish processing the document, which is the last date that I make edits to the centerlines pertaining to the document. Some documents take more than one day to finish, so this is why I list the last day.

The miscellaneous division contains one folder for each of the following categories:

Jurisdiction
Speed Limits
Weight Limits
Street Name Changes
Street Names
One-Way Streets

Documentation pertaining to each category goes in its respective folder. These documents are also labeled with the basemap number and the date, but because there are not as many of these documents, they are not separated by basemap number. However, as time goes by, if I accumulate enough of these kinds of documents, I might separate them as well.

SOME COMMON OPERATIONS

Populating Centerline Attributes

When populating the centerline attributes, it is important to keep in mind that the centerlines are tied to our Master Address Database (MAD). This means you have to be careful when editing the attributes, because the person doing this has the ability to get the two databases (SDE and MAD) out of sync.

Eighteen centerline attributes exist in the MAD, the rest don't. They are:

CENTERLINE_TAG
L_ADD_FROM
L_ADD_TO
R_ADD_FROM
R_ADD_TO
L_ZIP
R_ZIP
COUNTY_LEFT
COUNTY_RIGHT
CITY_LEFT
CITY_RIGHT
TWP_LEFT
TWP_RIGHT
PRE_DIR
STREET_NAME
STREET_TYPE
SUF_DIR
FULL_STNAME

Anytime any of these attributes is revised, the *CMFI* ("Custom MAD Feature Inspector") must be used. This is because the *CMFI* is linked to the MAD, whereas the ESRI attribute inspector window is not. Changes made to any of these fields through the *CMFI* are propagated throughout the MAD, whereas if the ESRI Attribute inspector is used, they are not. Changes made to any of the other centerline attributes may safely be made with the generic attribute inspector. Either tool will make changes to SDE (the centerlines), but only the *CMFI* will also propagate the changes through the MAD. The *CMFI* is shown in Figure 90 on the next page.

You may wonder why I don't always simply use the *CMFI*, so I don't have to worry about this problem. The answer is there are advantages to using the generic attribute inspector whenever possible. First of all, when you enter an attribute value in the generic window, and hit <Enter>, the focus moves to the next attribute. In the *CMFI*, the focus remains on the current attribute. In order to move to the next attribute, you must either use the <Down Arrow> key and then hit <Enter>, or else click on the next attribute with the mouse.

To illustrate how inconvenient this is, here is the series of steps you must go through when entering consecutive attributes in the *CMFI*, assuming you don't want to use the mouse (which I usually don't). First you press <Enter> when the attribute you want to edit is selected — this makes the value editable. You type in the value you want, and then hit <Enter> again. This saves the value (to memory only). Then you press <Down Arrow> to go to the next field (or multiple times to get to the field you want), hit <Enter> to make that field editable, type in your value, hit <Enter> again to save the value, and so on, all the way down the list of attributes. So you can see there is quite a bit more work involved with using the *CMFI* compared to the generic tool. I asked our consultant to see if they could replicate the behavior of the generic tool, eliminating all the extra presses of the <Enter> key, but they were unable to find a way to do it. (In their defense, this may have been a limitation of the .NET programming environment.)

Figure 90 - The Custom MAD Feature Inspector (CMFI).

Another issue with the *CMFI* is that you can't reliably do global updates to multiple centerlines, like you can with the generic ESRI tool. This is because there is apparently a bug in the code that checks for invalid attribute entries, that causes various error messages to appear under certain conditions, and prevents the changes from being effected. So when using the *CMFI*, you are limited to changing values on one centerline at a time. (You can have multiple centerlines selected, but you must change each one individually.) Again, the consultant attempted to fix this problem, but was unable to.

There is a workaround that helps with both the situations described above. I discovered that I can use the generic Attribute window to make all my attribute changes, either globally or singularly, and then, before saving my changes, switch to the *CMFI*, and then apply the changes to the centerlines one at a time. Finally, save the changes. This is why I like to have both windows open at all times, because I

frequently switch back and forth between them to accomplish the attribute edits.

If I have a lot of edits to make, I will usually do them in the ESRI editor, then switch to the *CMFI*, and apply the changes. If I only have a few changes to make, then I might do them exclusively in the *CMFI*. If I am only making changes to attributes that don't participate in the MAD, then of course there would be no need to use the *CMFI* at all.

Using the *CMFI*

The *CMFI* is one of the custom tools that make up the "Centerline MAD Tools," and is activated by pressing the next-to-last button on the Centerline MAD Tools toolbar:

The first time you click one of the Centerline MAD Tools, you will be prompted to log in to the application:

If one or more centerlines had been selected when the *CMFI* button was pressed, their attributes will be present in the *CMFI* window when it appears. It works just like the generic ESRI tool when multiple centerlines are selected; pointers to all the centerlines are listed in on the left side of the window, and the attributes are listed on the right. I always use the FULL_STNAME field as the field to be labeled in the "tree" on the left side of the window. If no centerlines are selected, the *CMFI* will of course be blank.

When one or more different centerlines are then selected, you must press the Refresh button on the *CMFI* in order for the new attributes to be loaded; they are not automatically loaded, like in the ESRI Attribute tool.

When new attribute values are entered, or existing values are revised, you must press the Apply button in order for the changes to be propagated through the Master Address Database. Pressing Apply doesn't *save* the changes, but it initiates a series of SQL commands that, when committed by invoking the ESRI Save command, actually update the appropriate MAD tables. It is important to remember that, just like with the ESRI attribute tool, no changes are committed until the "Save" command is executed. Each change that is "applied" doesn't need to be saved immediately. Just like with the ESRI editor, all changes that have been made since the last Save will be saved the next time the Save command is used.

If multiple centerline attributes are being updated with the *CMFI*, then the Apply button must be pressed after editing *each* centerline's attributes. The Apply button only causes changes to be made to the attributes that are visible in the *CMFI* window — whichever centerline currently has the focus.

The Revert button is used to restore the attribute values back to what they were before changes were made. However, if changes have been saved, you cannot revert back to their old values.

The Close and Help buttons are self-explanatory. If I need to use the *CMFI* during an edit session, I typically leave it open until I'm ready to end the edit session, in case I need it for more edits.

There is some error-checking code in the *CMFI* that checks for various invalid attribute conditions. For

example, there are a few fields that must not be empty, including STREET_NAME and FULL_STNAME. Another example is the value you enter in the FULL_STNAME field must already exist in the MAD as a valid street name. The error conditions are checked with you click the Apply button. If any errors are present, the offending fields are highlighted in red, and the changes are not applied. At this point, you can change the values that are invalid and then reapply the changes.

A nice feature of the *CMFI* is that if you make changes to a feature's attributes, and then attempt to select another feature without applying your changes, you will get a message stating there are pending changes on the form, and you are asked if you want to apply those changes before loading the attributes of the next feature. This is very helpful, since I occasionally forget to apply my changes.

Automatic Populating of Attributes

One of the nice features of the Centerline MAD Tools Extension is that certain attributes are automatically populated when a new centerline is created. This feature works with either the ESRI attribute inspector or the *CMFI*, as long as you are logged into the MAD Tools beforehand. The attributes that are populated are the ones which represent a geographic area (polygon). They are:

L_ZIP
R_ZIP
L_TRACT
R_TRACT
COUNTY_LEFT
COUNTY_RIGHT
CITY_LEFT
CITY_RIGHT
TWP_LEFT
TWP_RIGHT

The way this works is that the attributes are populated from the field of each corresponding layer that is used for labeling. The layers must be present in your ArcMap Map Document. The MAD.dll.config file, which is a part of the MAD Tools installation, contains the information that maps the respective layers (with their label field) to their corresponding attributes. Theoretically, if more boundary-related attributes are added to the centerlines in the future, they could also be automatically populated, as long as a layer were available that contains the boundaries, and the corresponding entry were added to the config file.

All the attributes that are automatically populated derive their values from the same layers that I use for display purposes, except two, the CITY_LEFT and CITY_RIGHT attributes. I had to create a custom layer for these values, because we did not already have one that was sufficient. Our *Exclcity* layer only contains the boundaries of the four excluded cities, but in order for the layer to work with the MAD Tools, I needed a layer that would include the geographic extent of all the area for which we have centerlines. I basically combined our *Exclcity* layer with the nine-county *Boundary* layer to achieve what was needed. I called this layer simply *Cities*, and it is pictured in Figure 91 on the next page.

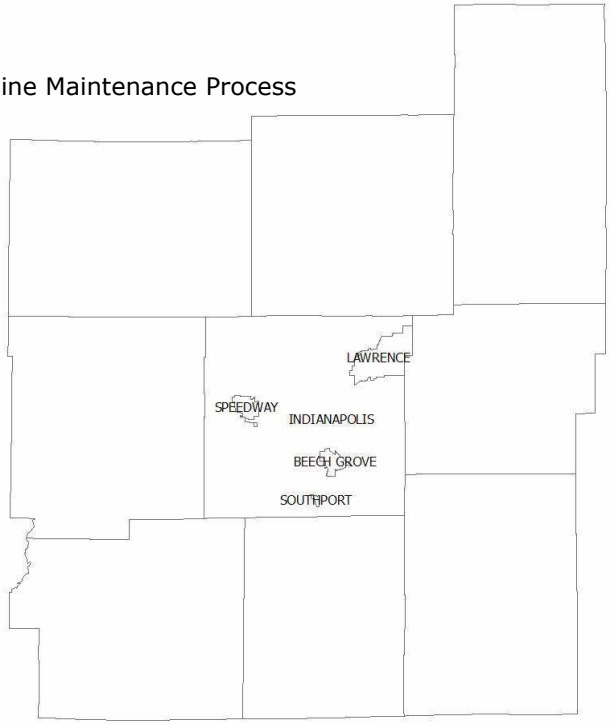


Figure 91 - The *Cities* layer.

This layer enables any centerlines outside the excluded cities but within Marion County to be populated with the value of "INDIANAPOLIS." Anything outside Marion County will retain null values.

One thing I have to be careful of is that if I digitize a new centerline that crosses one of the boundaries for one of the attributes, those attributes won't get populated automatically. Theoretically, there aren't supposed to be any centerlines that cross any of the geographic boundaries without being split, but there are exceptions, where the distance between an intersection and the spot where the centerline should be split is short. In these cases, I have to populate the missing attributes manually.

There is a way to automatically calculate the attributes of existing centerlines. This is what the Bulk Calc button on the MAD Tools toolbar does. It works on whatever centerlines are selected. Again, if any centerline crosses a geographic boundary, the attributes for that boundary won't get populated.

Digitizing a Street Centerline

Tangents

The easiest case is where a street has nice, uniform right-of-way to follow. In this case, I use the Midpoint Tool on the Sketch Tool dropdown and click on both sides of the right-of-way to create the centerline. This is similar to the Centerline Method of the old ArcInfo Add command.

If a street goes through an intersection, and it looks like the right-of-way for the street lines up on both sides of the intersection, I will ignore the intersection, and pick beginning and ending points for the centerline on either side of the intersection, as if the intersection didn't exist. In this case, I am assuming that the street alignment doesn't change direction through the intersection. Without knowing the exact angles of the right-of-way segments making up the intersection, it is impossible to know if there is a slight change of direction or not. Therefore, this assumption is subject to some error. However, I feel this is an acceptable risk, because our centerlines are not intended to be "survey" accurate. My rule of thumb when creating centerlines that go through an intersection is, if the street *looks* like it goes straight through the intersection, then that's the assumption I go on.

Some streets have bends in them, but still consist solely of tangents. In this case, I am careful to only

click on the right-of-way where it bends, so that I don't introduce superfluous vertices into the centerline.

If the street happens to follow a section line, then I will do a copy and paste from the *Seclines* layer to the *Streets* layer, and use that for the centerline.

Streets without right-of-way are a little harder. If a street has curbs on both sides, I will use the Midpoint Tool to place the centerline between the curbs (on the aerial photos). If the street has clearly delineated edges of pavement, I will use the edges. Otherwise, I will just eyeball the centerline. In apartment complexes where there are parking spaces on one or both sides of the centerline, I place the centerline in the center of the traveled way. See Figure 92 below.

Sometimes, if part of a straight stretch of street has clearly defined edges, and part doesn't, I will digitize the centerline in the area that has good edges, and then use the ESRI Scale tool to stretch the centerline to make it as long as it needs to be.

Curves

Almost all the curved streets we have are composed of one or more circular curves. I believe this is due to the simplicity for developers of designing circular curves, as compared to ellipses, spiral curves, etc. This, of course, makes it easier to digitize centerlines for these curves.

You might ask, "How do you know if a curve follows a circular path, or some other type of curve?" The answer is the same as with straight streets. If the curve *looks* like a circular curve, then it *is* a circular curve, as far as I'm concerned. If a curve appears to follow an elliptical path for a significant length, then I will use the Ellipse Tool on the Line Edit Toolbar to construct an ellipse of the right shape (using the Rotate and Scale Tools), and clip the portion that I need. If the curve is irregular, I will either digitize it as a series of short circular arcs, or create points along the centerline using the Midpoint Tool, and then smooth the curve using the custom Smooth Tool.

Figure 92 - Apartment centerline.

Creating a Three-Point Curve

This process is based on the simple fact of geometry that three points define a unique curve. This process is used most frequently for streets with circular rights-of-way, but can also be used for other streets, where the edges are well defined. The Midpoint Tool is ideal for creating these types of curves.

When using the three-point method to construct a circular arc, two of the points must be the endpoints of the arc, with the remaining point lying somewhere in between. In order to locate the ends, I need to locate the points of tangency on the right-of-way lines. To do this, I turn on the display of the vertices on the *Parcels* layer by pressing the <V> key. Using the Midpoint Tool, I snap to the point of tangency on one side of the right-of-way, and then do the same for the same point on the opposite side of the right-of-way.

To locate a point in the middle of the centerline, I select a spot on the right-of-way approximately halfway between the two ends of the curve, snap to a vertex (or edge if the right-of-way curve is a true curve), and then snap to another vertex (or edge) that looks to be perpendicular to the first point. Although any set of points between the endpoints theoretically would work, selecting a point approximately halfway between the endpoints helps to minimize any random error inherent in the right-of-way when it was digitized.

Finally, the second endpoint of the curve is located in the same manner as the first endpoint. The result is a temporary line, bent in the middle, whose endpoints and single vertex represent three points on the curve. Then it is a simple matter to use the ESRI Arc Tool to create the curve by snapping to those three points, and then deleting the temporary centerline. See Figures 93 through 95 on the next two pages.

When creating centerlines for curved rights-of-way, I sometimes find that the curves are not tangent to the adjacent straight segments. Sometimes the deviation angle is small, and is only apparent when you zoom in closely. Sometimes the deviation is not so small, and easily visible to the naked eye. This can make for centerlines that don't look just right, because the curves don't appear to line up with the straight sections; nevertheless, if this is the way the right-of-way was platted, that is the way the centerline will be digitized.

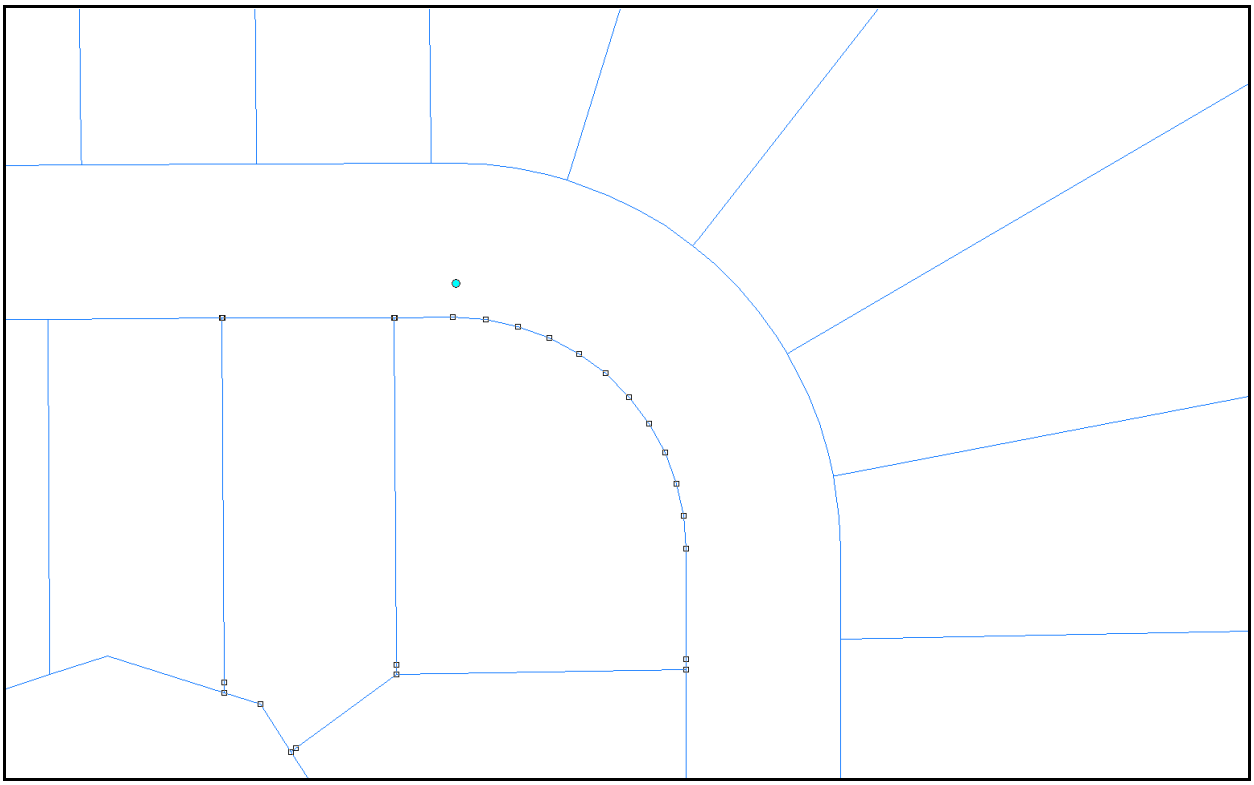


Figure 93 - Locating the point of tangency on the right-of-way to begin constructing a three-point curve.

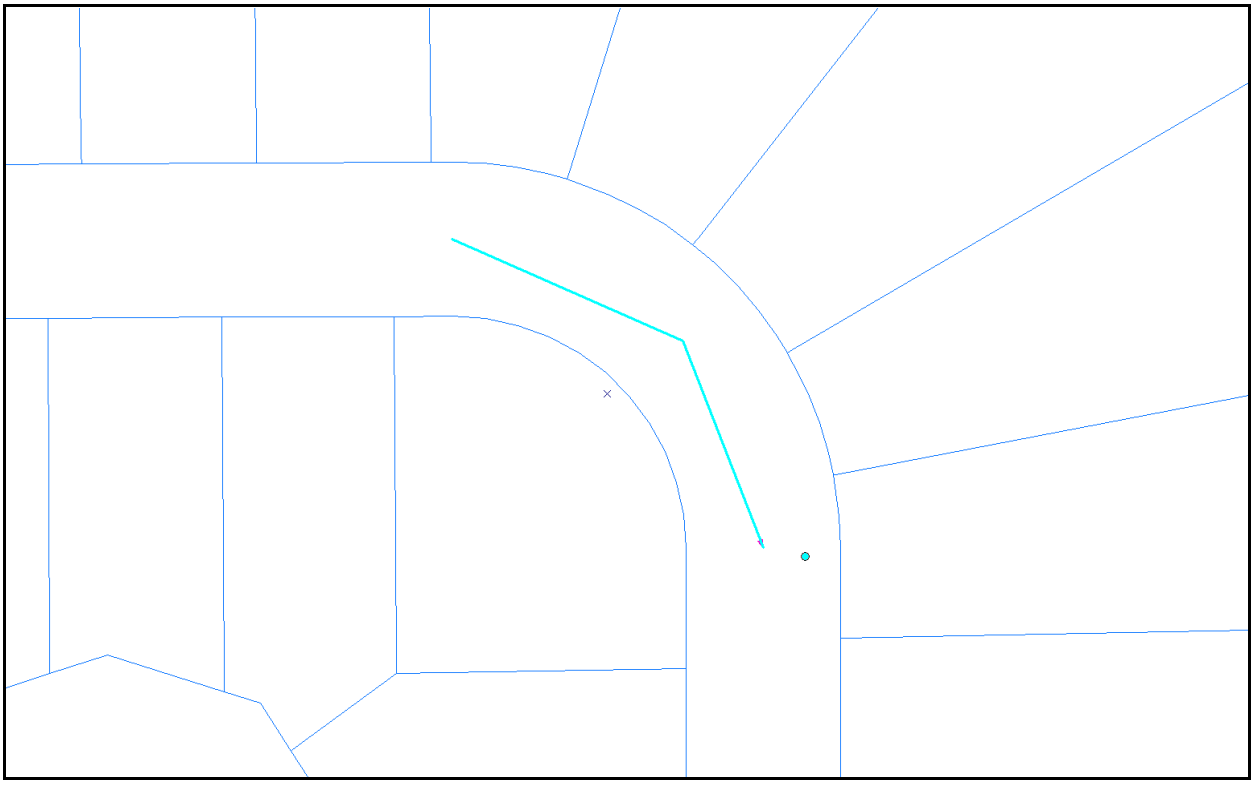


Figure 94 - The temporary centerline has been created, locating the three points of the new curve.

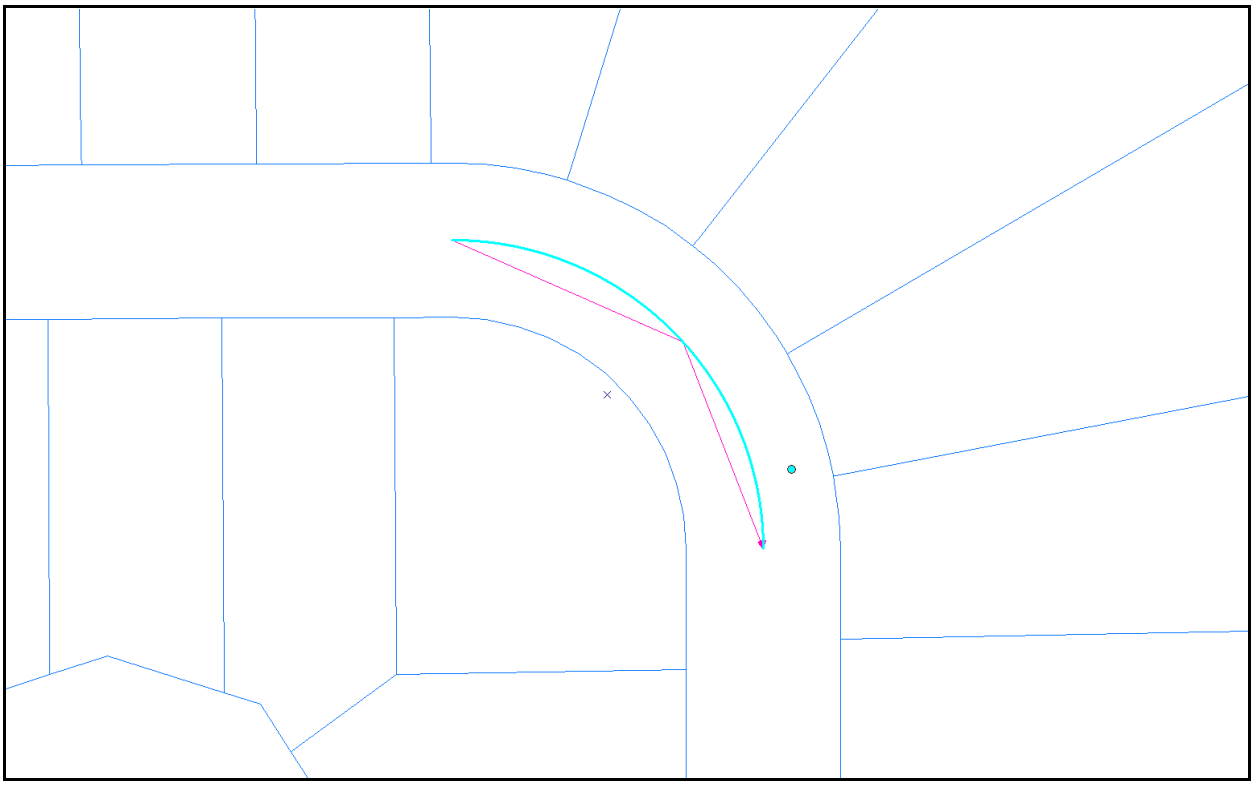


Figure 95 - The three-point curve has been created. All that remains is to delete the temporary centerline.

Creating Curves With the Fillet Tools

Sometimes when I'm digitizing, the situation calls for a circular (or nearly circular) curve, but there is no right-of-way or other reference that defines the exact alignment of the curve. In this case, I prefer to use one of the fillet tools, as opposed to free-handing the curve. That's because I prefer the appearance of an exact, geometric curve.

Using the Generic ESRI Fillet Tool

In some situations, I use the generic ESRI Fillet Tool. This tool lets you create a fillet by interactively varying the radius until it looks right. One good use for this tool is for creating relatively small-radius fillets on private streets, where the fillet radius does not have to match that of any other fillet. See Figure 96 on the next page for an example.

The fillet tools require that you have two intersecting tangents in order to create a fillet. In this example, one of the tangents has already been created. To create the other one, I zoom in to the portion of the street which is straight, and use the ESRI Midpoint Tool, using the edge of pavement as a guide. (Figure 97, next page.)

Next I use the Line Intersection Tool to extend the tangents for one of the curves to their intersection point. (Note that this process introduces two pseudo nodes, which must be removed, not only because they are superfluous, but because they may interfere with the filleting process.) See Figure 98, two pages over.

Finally, I zoom in to the area of the curve and use the Fillet Tool to interactively draw a fillet that looks like it follows the center of pavement through the curve. See Figure 99 for the result.

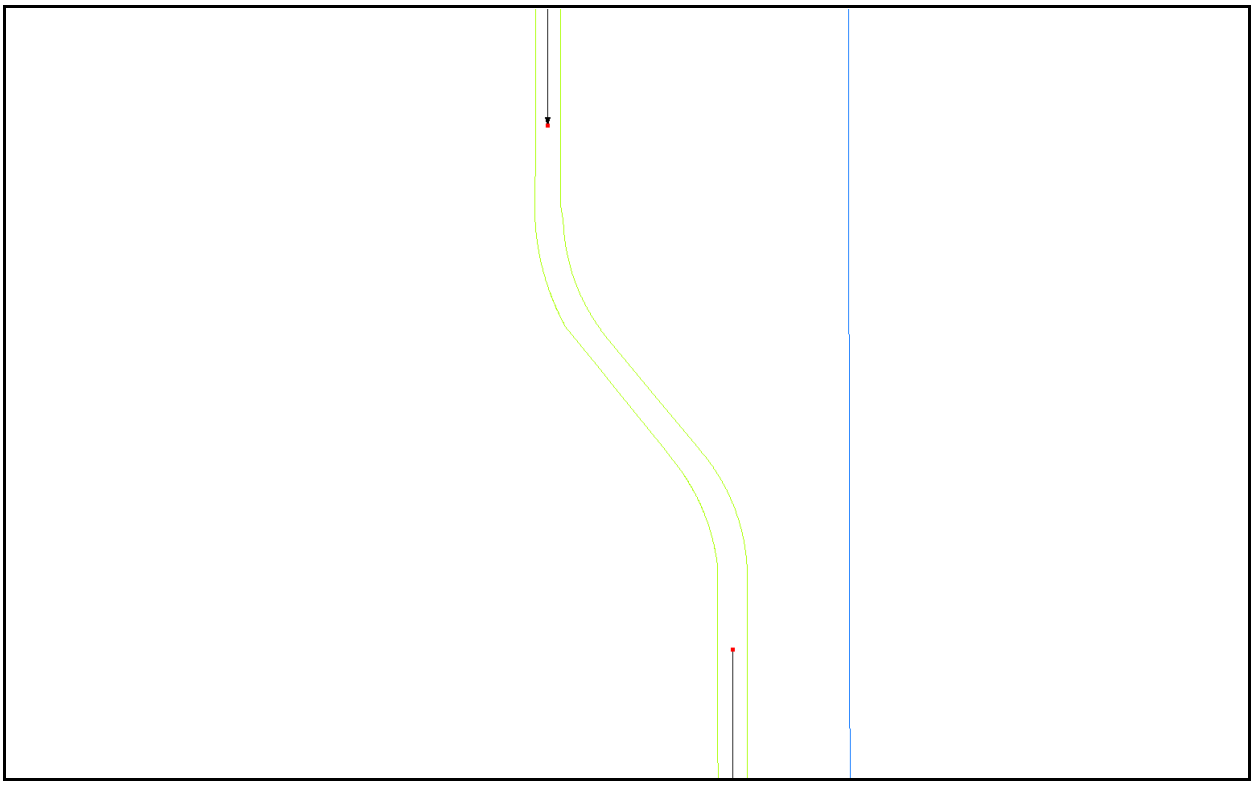


Figure 96 - A street centerline where the generic Fillet Tool is appropriate for creating the curves. Note the absence of right-of-way.

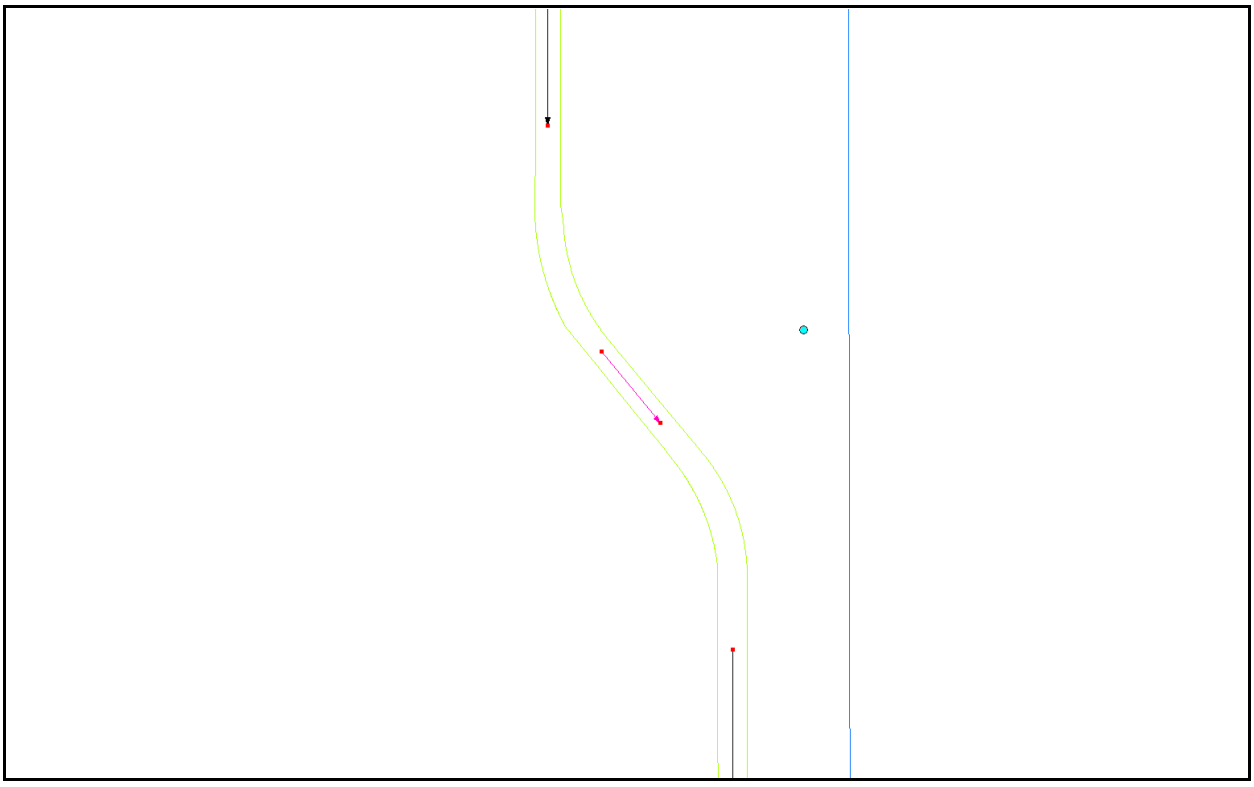


Figure 97 - A tangent centerline is created with the Midpoint Tool.

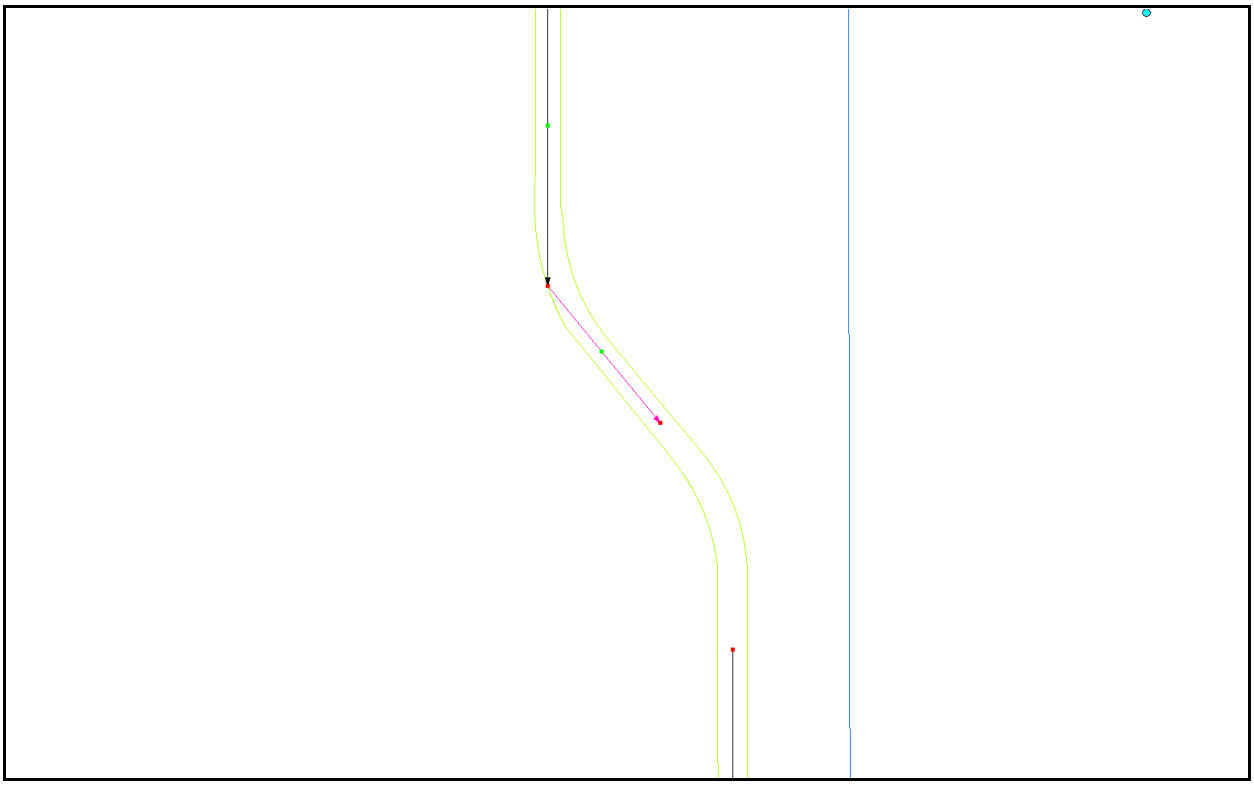


Figure 98 - Two tangents are extended to their intersection point.

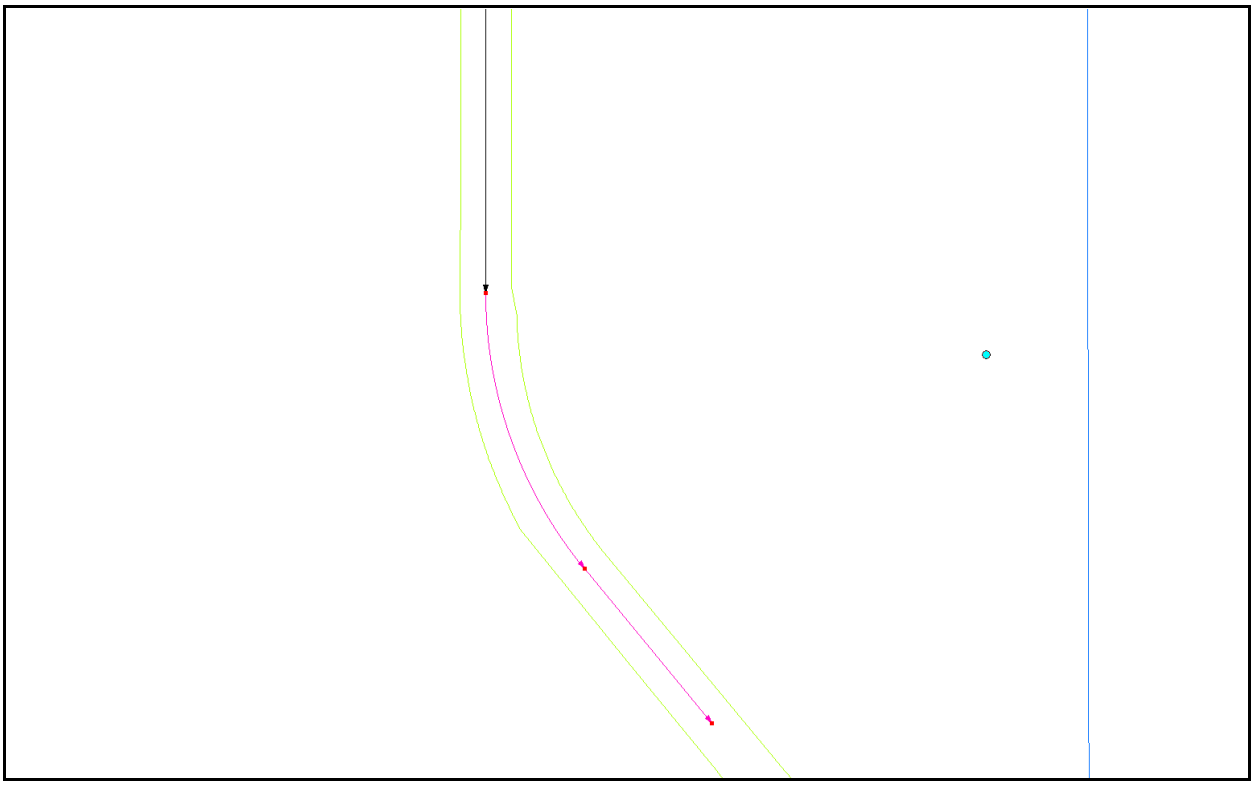


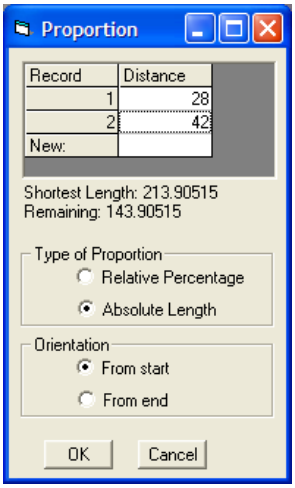
Figure 99 - One of the curves is complete.

Figures 101 through 106 on the next three pages show another example of the use of the generic ESRI Fillet Tool. In this example, I am digitizing centerlines in an apartment complex, and there is an area of pavement the centerline of which appears to follow a circular path. The arc appears to begin in the southwest portion of the image in Figure 101, and continues northeast, all the way to the circle.

In order to construct a fillet for the circular arc, I first need two intersecting tangents. The first tangent is already constructed. It is the centerline approaching from the southwest. Since I want the fillet to connect to the circle, I next draw a temporary centerline in a southerly direction that starts at the circle. The direction of the line will be the direction of the arc at the point where it intersects the circle. (I eyeball this.) See Figure 102.

Next I extend the centerline of interest (the north-south centerline) till it intersects the line I just drew, and split the new centerline at that point (Figure 103). Then I delete the portion of the new centerline that I don't need.

Now I am ready to take advantage of one of the tools on the Line Edit Toolbar called the Proportion Line Features tool. (It is the sixth tool from the left, in the picture of the Line Edit Toolbar on page 106.) Clicking on its button brings up the dialog shown here:



What I need to do is split my centerline of interest at a point which is the same distance from its endpoint as the length of the new centerline I just created from the circle to the point of intersection. First I select the existing centerline, and read the length of it at the bottom of the ESRI attribute inspector. Then I select the centerline that I want to split, click on the Proportion Line Features tool, and fill in the appropriate distance values. Distance 1 is equal to the length of the existing centerline I selected earlier, and Distance 2 is the remainder of the distance of the centerline. I select the "Absolute Length" and "From Start" options, and click "OK."

Now I have two equal-length centerlines that will form the basis for the curve I want to construct. I select them both (Figure 104), and zoom to them, using the ESRI Zoom to Selected Features tool. (Zooming in allows me to construct the curve more accurately.) Then I select the ESRI Fillet Tool and eyeball the appropriate curve I want (Figure 105). After the curve is placed, all the temporary left over centerlines are deleted, the new curve is merged with the rest of the centerline, and the result looks like Figure 106.

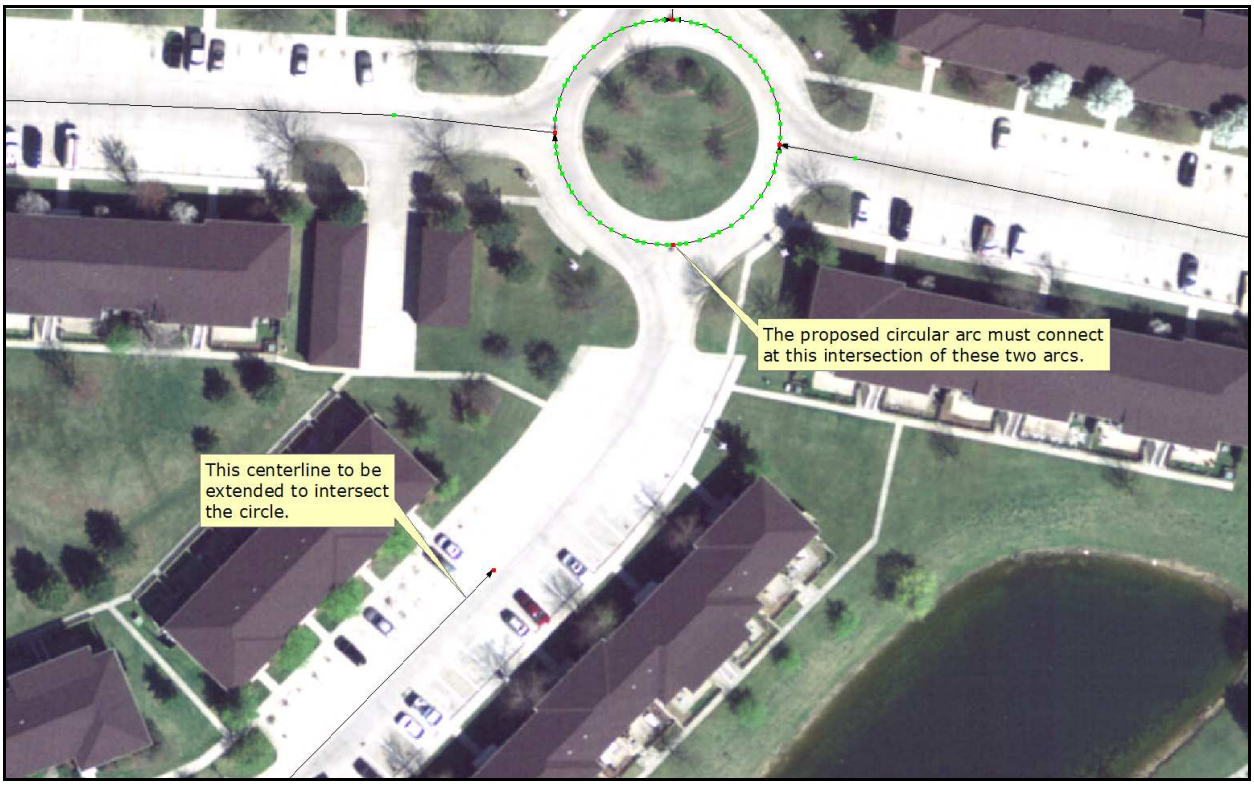


Figure 101 - Getting ready to tie a tangent centerline to a circular centerline, using a fillet.

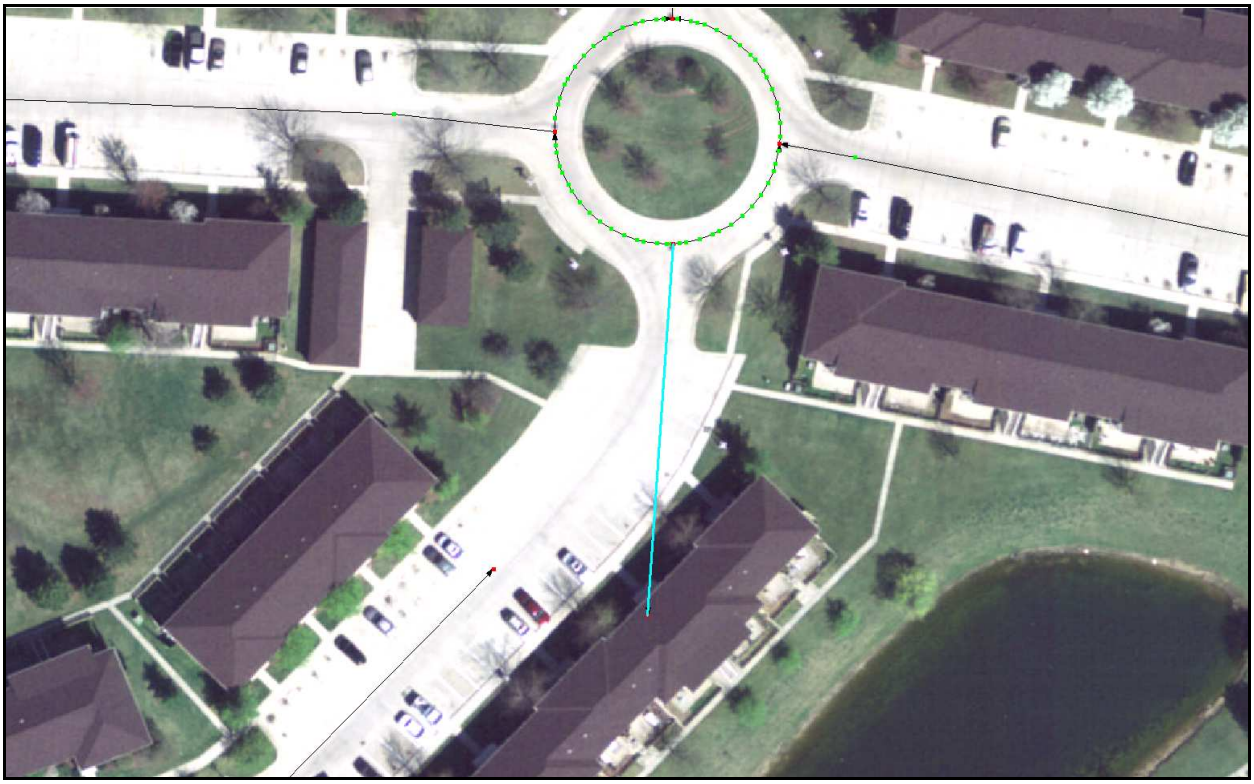


Figure 102 - A temporary centerline is constructed that begins at the circle. The direction of the line matches the direction of the proposed arc at the point where it will intersect the circle.

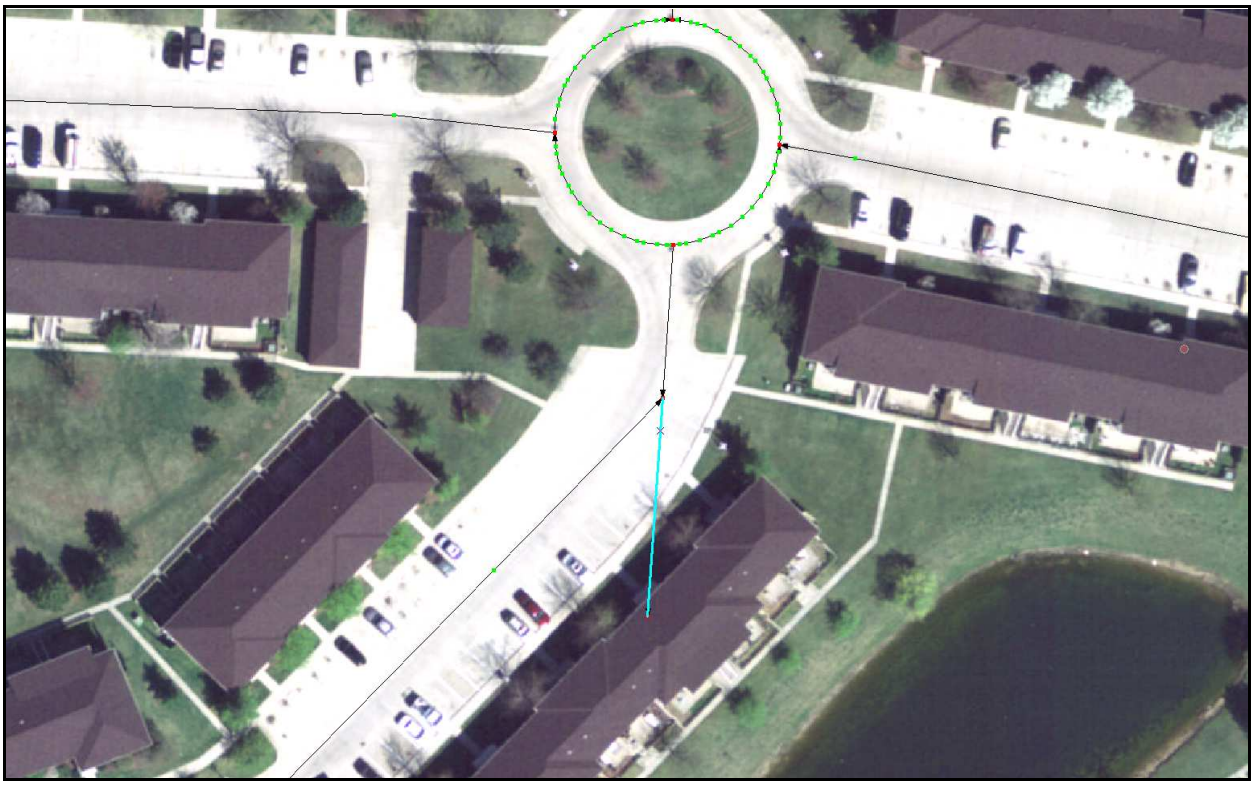


Figure 103 - The centerline of the north-south street is extended to intersect the new centerline, and the new centerline split at that point.

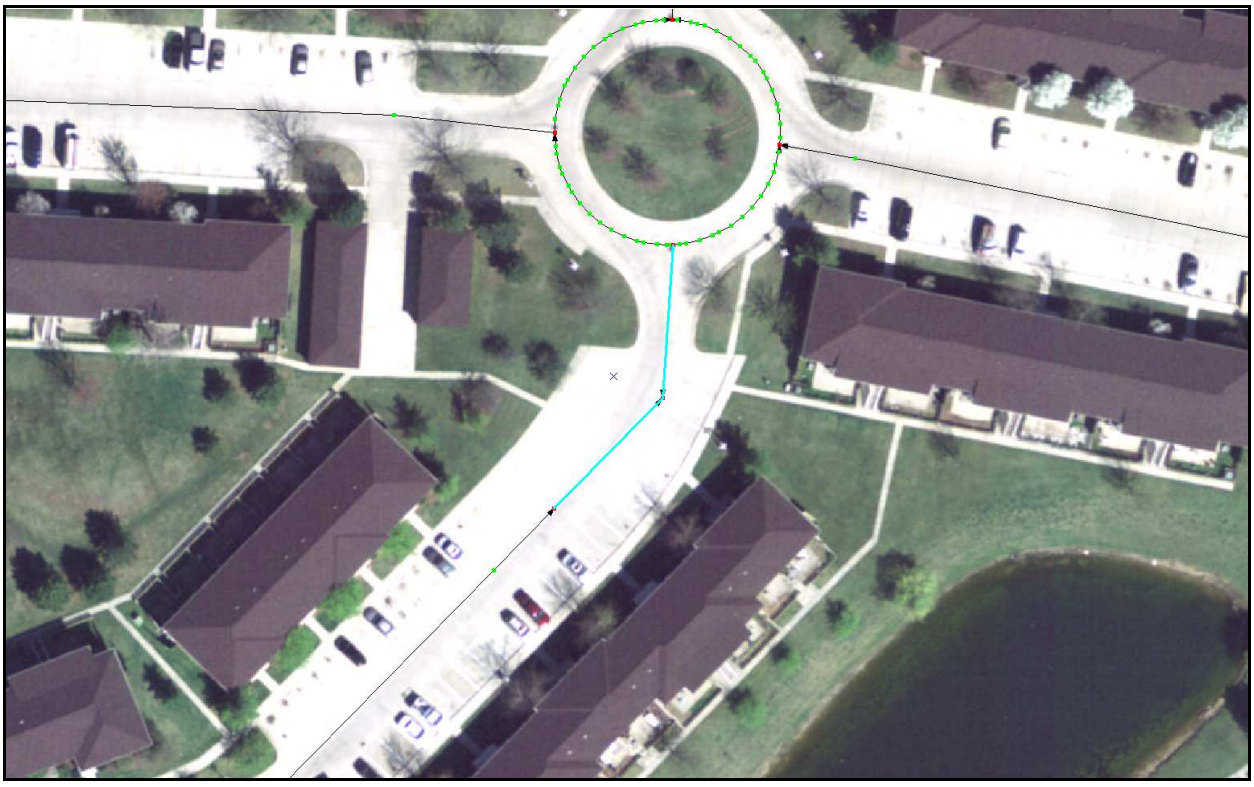


Figure 104 - The north-south centerline is split at a distance equal to the length of the new centerline.

Figure 105 - Constructing the desired curve, using the two tangent centerlines as a guide.

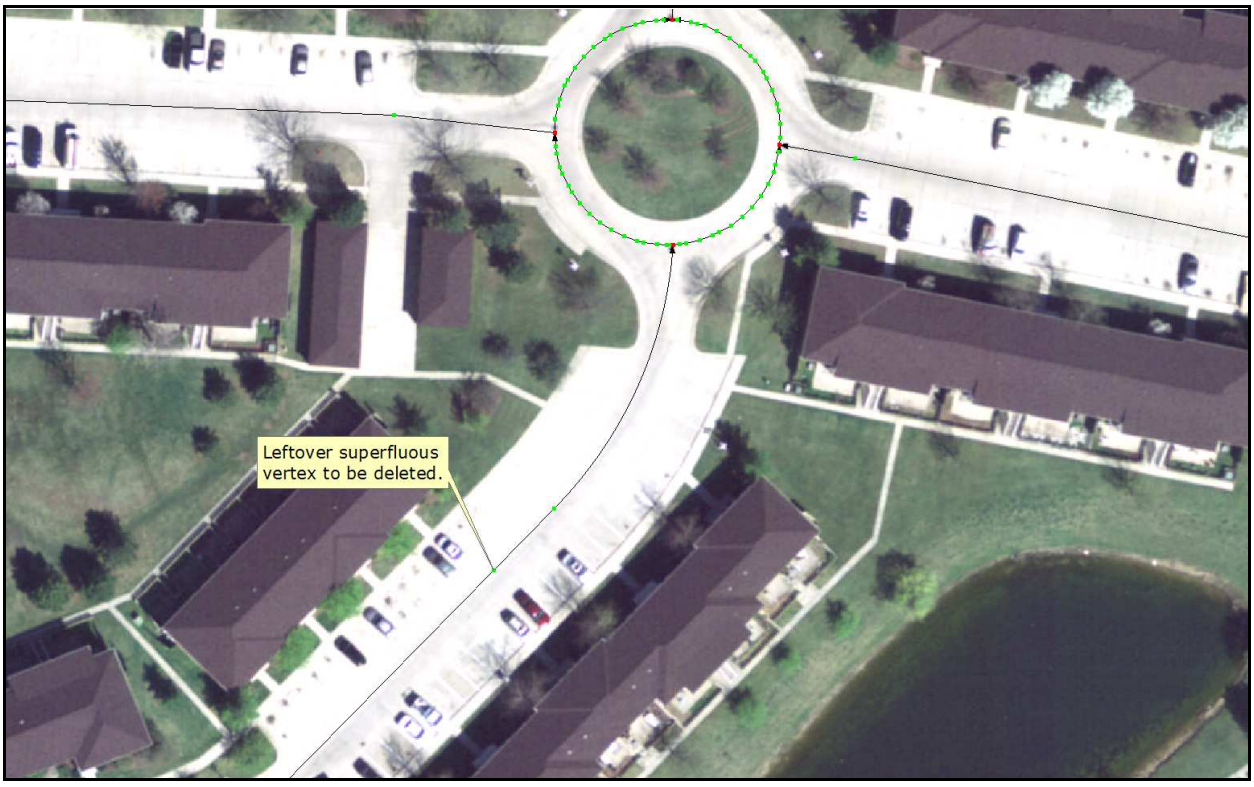


Figure 106 - The final result.

Using the Custom Fillet Tool

As explained on page 107, the first button on this tool allows you to create a fillet of a given radius (in map units) between two intersecting centerlines. One place I like to use this tool is for placing large curves on highway and interstate centerlines. The ESRI tool doesn't work well for these because you have to zoom out so far in order to be able to drag the cursor far enough to construct the large curve, that it is hard to see if the curve is in the center of the pavement. Figures 107 through 112 on the next three pages illustrate how I constructed one of these large radius curves.

The first image shows a portion of Interstate 465 that contains a very large radius curve. In the case of a divided roadway such as this, I digitize the centerline for one set of lanes, and then create the centerline for the other set of lanes as an offset from the first centerline. In this example, I decide to construct the centerline for the westbound lanes first.

Once again, in order to construct a fillet, I need a pair of intersecting tangents. To start, I zoom in to the westbound lanes, to the west of the arc. I pan to the west along this tangent portion of the interstate, away from the proposed arc, until I am quite a distance away from the arc. The reason for this is because the longer the tangent is, the more accurately aligned it is likely to be. It is important to try and get the tangents accurately placed, especially for large radius arcs, because the accuracy of the fillet is dependent upon the tangents used to construct it. I use the painted lane lines on the pavement as a guide to place the tangent endpoints. Figure 108 shows the first tangent after it has been digitized. (The scale of the photo is *not* the scale the centerline was digitized at — in the photo, I have zoomed out to show the entire centerline.)

Figure 109 shows both the tangents after they have been digitized. Now I extend and intersect the tangents, and remove the resulting pseudo nodes. Then I select the two tangents, and zoom in to the area where they intersect. Now I press the first button on the custom fillet tool, and click on the map near where the proposed fillet will be. A little box pops up, where I can type in the desired radius. At this point, I have to make an educated guess as to what radius will produce the desired curve. From experience, I know that 10,000 feet is a good beginning point for this curve, so I type that number in the box and hit <Enter>. The result is shown in Figure 110.

I notice that the resulting arc is on the outside of where I want it to be. I know intuitively that this means that I need an arc with a bigger radius. So I hit the <Undo> key, and repeat the process, using a radius of 20,000 feet. The result is shown in Figure 111. This arc is just a little inside where I want it to be, so I know I'm getting close. From here on, this is just a trial-and-error process, trying different radii, until I find the one that aligns perfectly with the centerline of the pavement. I finally hit upon 19,100 feet as the radius that produces a fillet of just the right size (Figure 112).

The final steps to complete the interstate centerlines involve merging the fillet with its two tangents, measuring the distance from this new centerline to the centerline of the eastbound lanes, and then using either the ESRI Trace Tool (with an offset) or Copy Parallel Tool to create the centerline for the eastbound lanes.

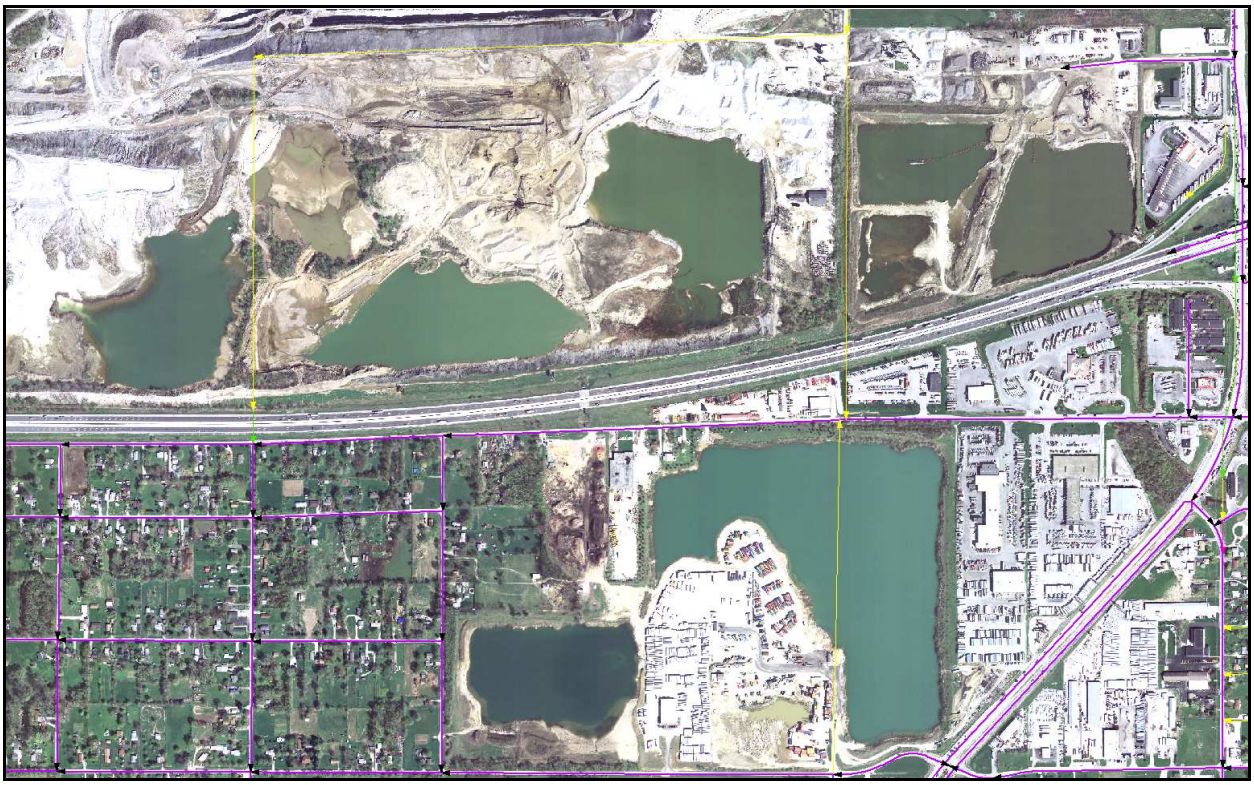


Figure 107 - Example of curves required of a very large radius.

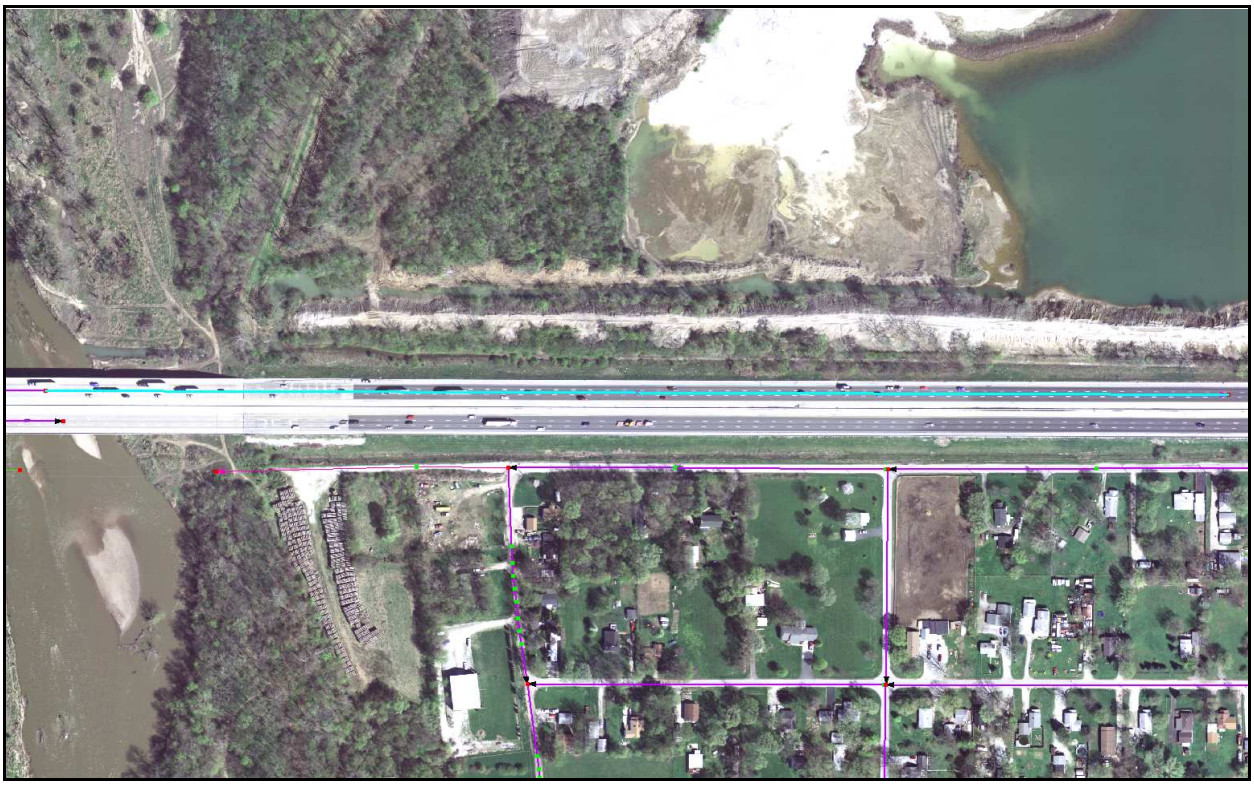


Figure 108 - The first tangent is digitized.

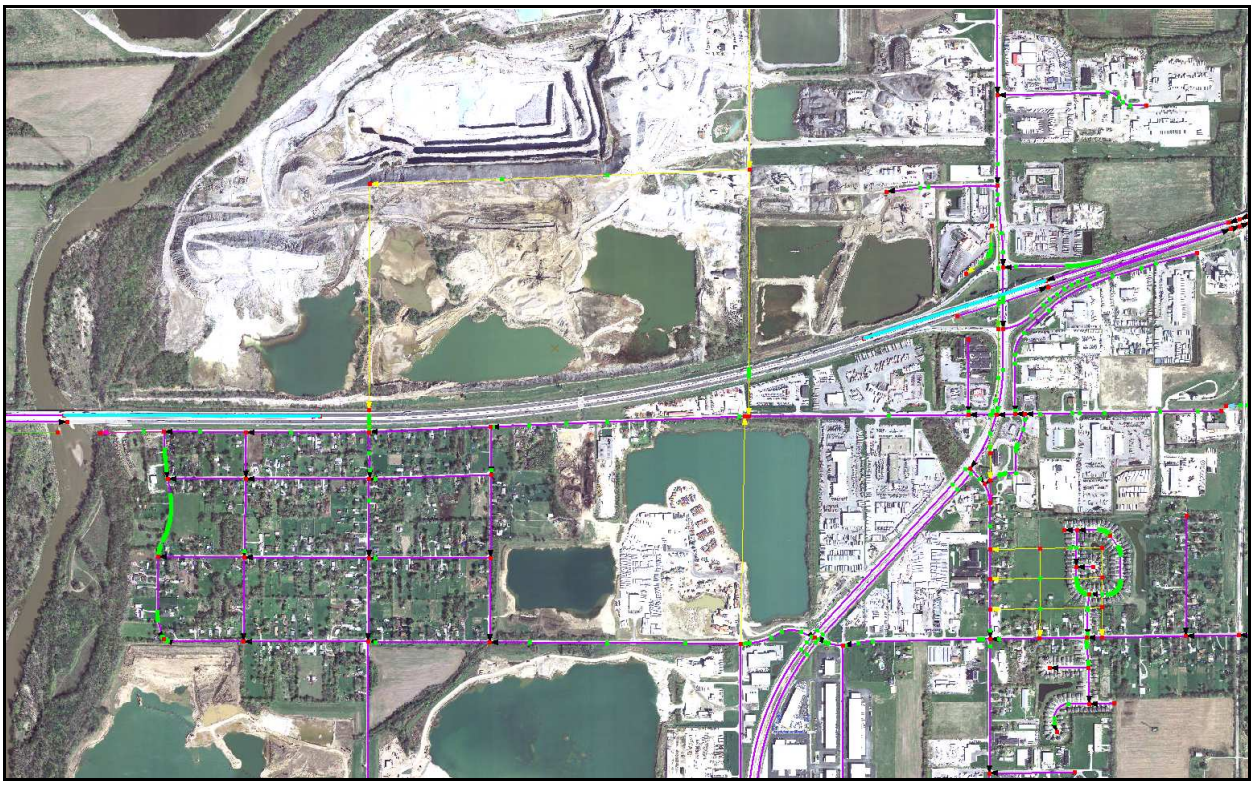


Figure 109 - Both tangents have been digitized.

Figure 110 - The first guess.

Figure 111 - The second guess.

Figure 112 - Just right!

The second button on the Custom Fillet Tool will fillet all corners of a given centerline with the same radius. The most common application for this is in apartment and condominium complexes, where a centerline makes a loop like the one in Figure 113, next page. Here, I want all the fillets on the loop to be the same size. Since the radii of the corners of the loop are somewhat arbitrary, I like to make them all identical, for best appearance.

I start by digitizing the tangents, intersecting them, and then joining them into one centerline. Then I press the "Fillet All Corners" button, and I type the desired radius. I guess on the radius on the first try; if I guess wrong, I simply undo the change, and then try a different radius until I find one that looks good. The result is shown in Figure 114.

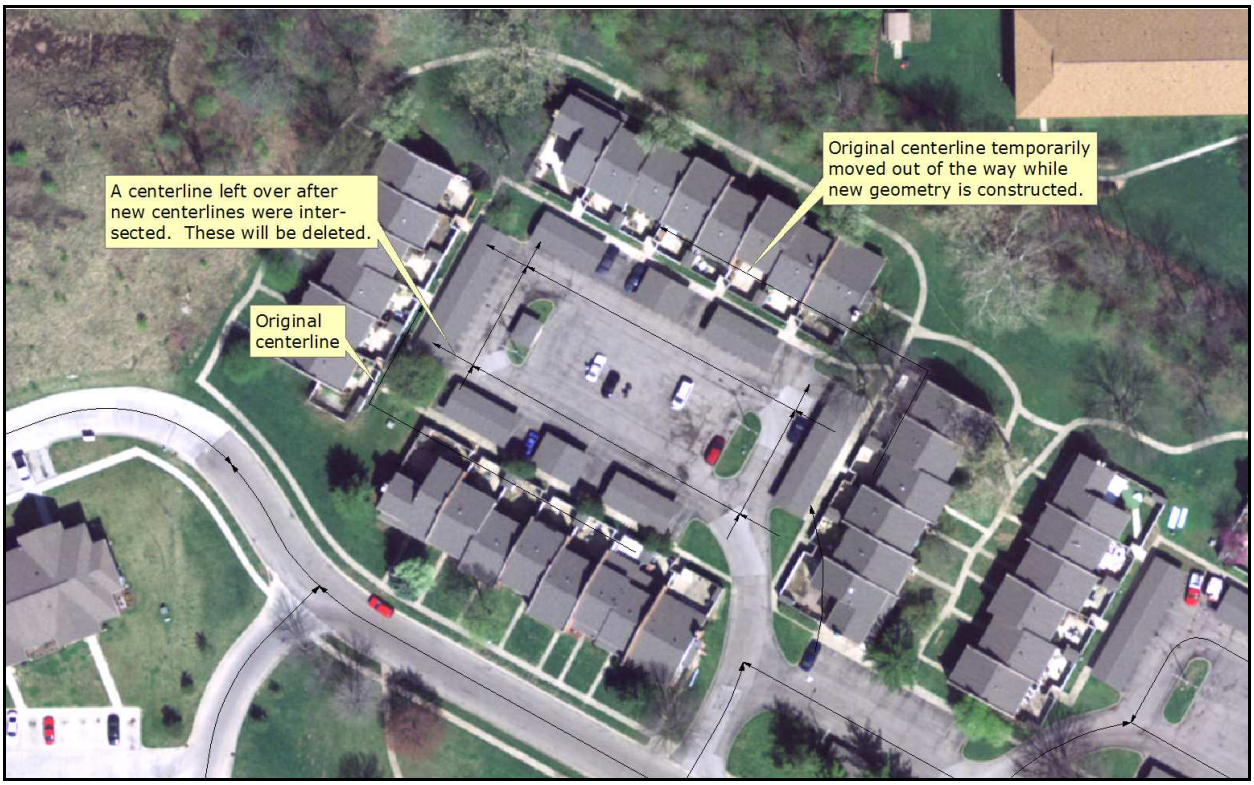


Figure 113 - Some rough apartment centerlines are being replaced with some that are more geometrically accurate. Here there are three fillets that I want to construct with the same radius.

Figure 114 - The result. Once I am finished cleaning up the new centerlines, the attributes will be transferred over from the old centerlines, and then the old centerlines deleted.

Splitting and Merging Centerlines With the Centerline MAD Tools

It is important to remember that as long as you're not affecting the attributes in the MAD, you can use the generic ESRI Split and Merge commands when editing the centerlines. The Split and Merge commands on the Centerline MAD Tools are only to be used when the actual address records associated with the centerlines are affected.

For example, I often decide to revise the centerlines for streets in order to improve their alignment. The attributes don't need to be changed. In this case, as in the example of the apartment centerlines shown in Figures 113 and 114, I will temporarily move the existing centerlines to the side, construct the new centerlines, and then transfer the attributes from the old centerlines to the new centerlines. While constructing the new centerlines, I may need to split and merge them many times, in order to arrive at the configuration that I want. I can use the generic commands to do this, because I know that the end result will have no effect on the MAD attributes of the centerlines. I am not creating or removing any intersections, therefore the end result is the same as the beginning, as far as the MAD is concerned. The only difference is that now the centerlines are shaped differently.

However, if I need to split a centerline because a new street has been created that intersects the centerline at a new intersection, then I need to use the custom Split command on the Centerline MAD Tools toolbar. This command not only splits the geometry of the centerline, it creates another address record in the MAD, so that there are now two records in the MAD that encompass the address range formerly occupied by only one centerline.

Splitting

The Centerline MAD Split command works similarly to the generic Split command, whereby you select the point on the centerline where you want to split the centerline. The difference is that the generic Split Tool only splits the geometry; it does not affect the MAD, whereas the MAD Split Tools splits the address records in the MAD as well as the geometry. Only one centerline at a time may be split. Since the place where you need to split the centerline is usually where another street is intersecting the centerline, you need to split the centerline at the exact point where the streets intersect. This means you must have the new side street already located before invoking the Split command, so you know the exact point to snap to, where the split should take place.

Consider the example in Figure 115, next page. In this example, the street highlighted in blue (Ridgegate West Drive) is a new street in a new subdivision received from Brian Schneider. This street will intersect 79th Street. Notice that the centerline Brian gave me extends slightly beyond 79th Street. This needs to be fixed before 79th Street can be split.

With the street selected, as it is here, I select the Split at Intersection Tool on the Line Edit Toolbar, and then click on the centerline for 79th Street. This splits the selected centerline exactly where it crosses 79th Street, without splitting the centerline for 79th Street itself. I delete the tiny dangling centerline for Ridgegate West Drive, and then switch to the Centerline MAD Split tool. Now I have an exact point to snap to in order to split 79th Street, the place where Ridgegate W. Dr. intersects the 79th Street centerline. If the centerline for Ridgegate W. Dr. had not intersected 79th Street to begin with, I would have extended its centerline until it touched the 79th St. centerline, again giving me the exact point where the centerline for 79th St. needed to be split.

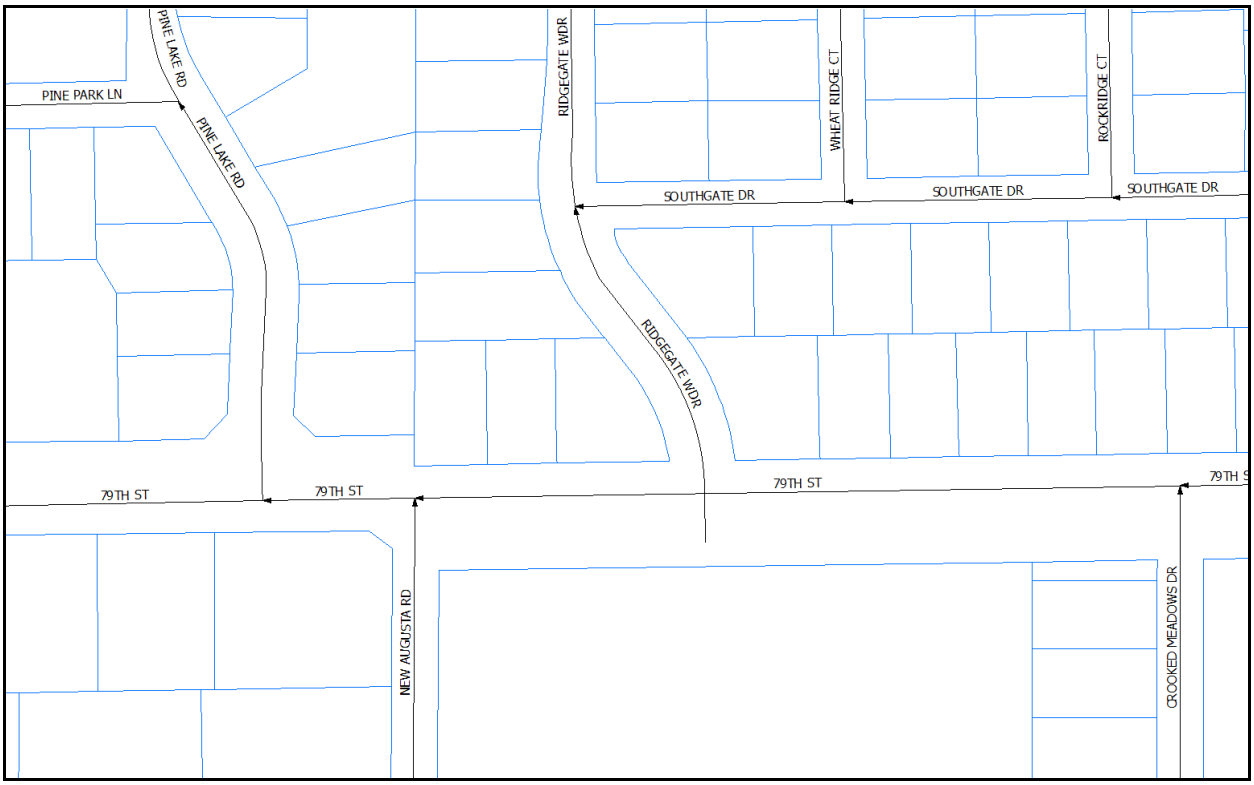


Figure 115 - 79th St. needs to be split where Ridgegate West Drive intersects it. Since a new intersection is being created, the Centerline MAD Split Tool must be used for the split.

Another occasion that comes up sometimes where I need to use the Split at Intersection Tool, before I can use the MAD Split Tool, happens when subdivisions are accepted by the City. Frequently subdivisions will be accepted a Section at a time, because this is how they are often platted and built. Usually Section boundaries do not coincide with intersections. This means street centerlines in subdivisions sometimes need to be split at Section boundaries, because the parts of the streets on either side of the boundaries will contain different acceptance dates.

Consider Figures 116 and 117 on the next page. Here we see part of two Sections of a subdivision called Wildcat Run. (The yellow line shows the boundaries of the Sections.) Suppose Section 3 has just been accepted, but Section 5 has not. This means Bracken Dr. needs to be split at the boundary between Sections 3 and 5. How to locate the exact point to apply the MAD Split Tool? First I will construct a temporary line across the right-of-way of Bracken Dr. that lines up exactly with the Section boundary. (I have snapping activated for the *Parcels* layer.) I can't simply use the Intersect Line Features or Planarize Lines Tools, because they don't add new records to the MAD. So, with the temporary line still selected, I use the Split at Intersection Tool to click somewhere on the centerline for Bracken Dr. This breaks the temporary line where it intersects Bracken Drive, as in Figure 117. Then I use the MAD Split Tool to split the Bracken Dr. centerline exactly at the point where the temporary line was split, by snapping to that point.

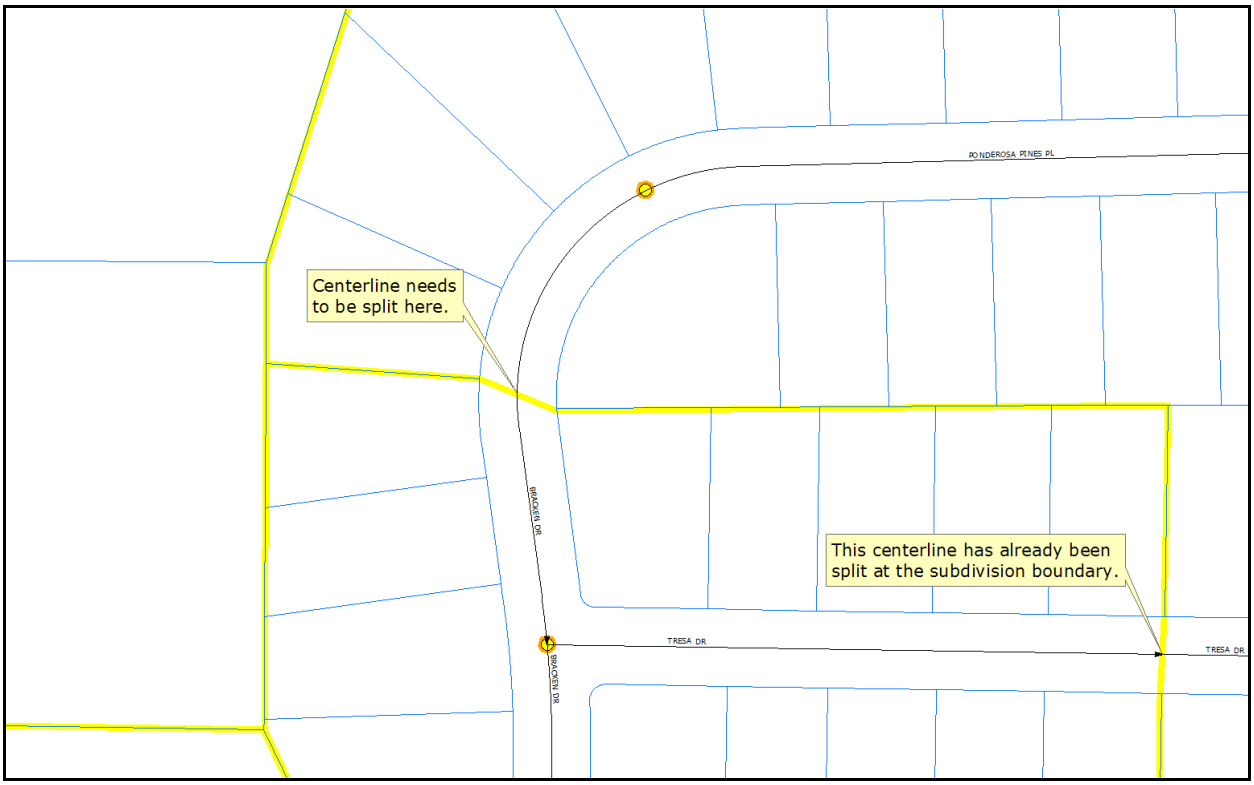


Figure 116 - A centerline that needs to be split at a subdivision section boundary.

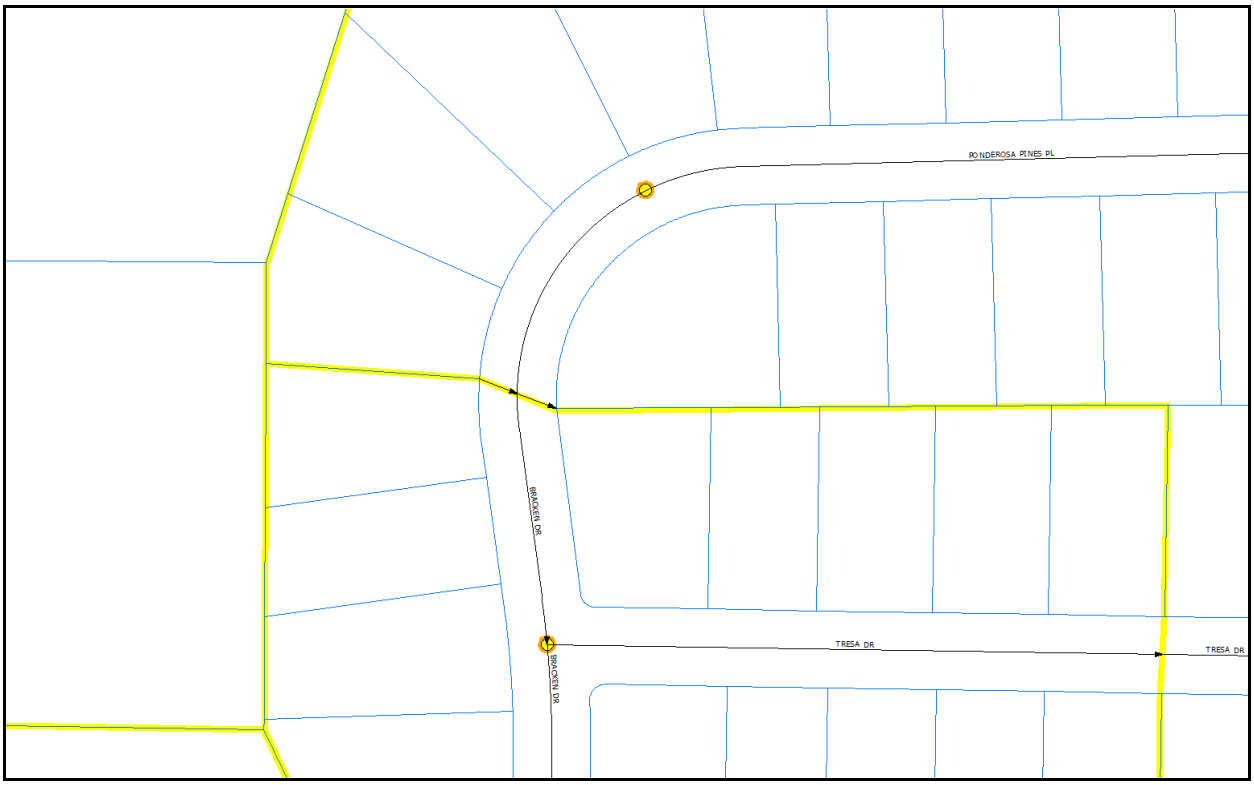


Figure 117 - A temporary line has been constructed and split where it intersects Bracken Drive. This give me a point to snap to, which is the exact point where Bracken Dr. needs to be split.

Once you click on the appropriate point with the MAD Split Tool where you want to split a centerline, the *CMFI* pops up (if it wasn't already displayed), containing the attributes of the two resulting segments after the initial one was split. Here you may edit any of the attributes. The address range of the initial segment has automatically been split, in proportion to the lengths of the resulting centerlines. If this is acceptable, no change needs to be made to the address ranges, but if not, you can change the resulting ranges to whatever they need to be. Also, at a minimum, the DATE_MOVED and DATE_CHANGED fields need to be populated with the current date, because splitting a centerline always constitutes a move, as well as a change to the attributes. (Exception: It is conceivable that I would want to keep exactly the same attributes on one of the two new centerlines as existed on the original centerline. In this case, the address range on the other centerline would be all zeros. For this exception, which doesn't happen very often but does happen, the DATE_CHANGED field on that centerline would not be updated.)

The Apply button needs to be pressed after making the changes to *each* new centerline. Before pressing the Apply button, one of the new centerlines resulting from the split will retain the Tag number of the initial centerline, as displayed in the *CMFI*, and the other one will be zero. When the Apply button is pressed for the one that has the zero Tag, a new Tag number will be automatically assigned to this centerline. The Tag number is drawn from a sequence table in the MAD that automatically keeps track of the last Tag number assigned, so that each new Tag number is always one greater than the last number assigned.

After the split, a new intersection point needs to be added to the *Intersection* layer. The new point is, of course, snapped to the actual intersection. Then the *CMFI* is refreshed, which now displays the attributes of the new intersection (which are all blank). The attributes are populated, and when the Apply button is pressed, a new consecutive Tag number for the intersection is assigned. Note that the sequence for intersections is separate from the one for centerlines.

The split is not committed until the changes are saved.

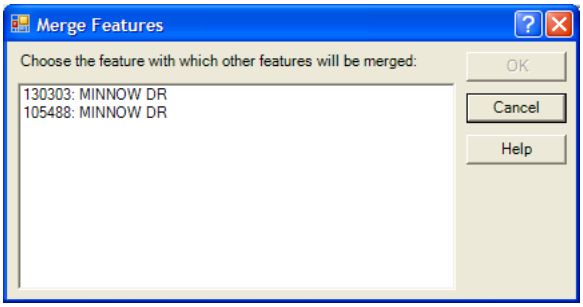
[Update for second revision: As we have updated our versions of ArcSDE and ArcMap over the last few years, the MAD Split tool has quit working properly. Therefore, I have to split the centerlines with the generic Split tool, and revise the resulting address ranges and other attributes manually. In the future, we may pay a consultant to update the MAD Split tool, so it will work with the newer versions of the software.]

Merging

Just like for splitting, it is important to remember that the Centerline MAD Merge Tool is only to be used when two address ranges need to be combined into one. (A physical intersection may be getting removed as well, but the logical intersection will remain, as they are never deleted.) Only two centerlines may be merged at a time, and they must be adjacent.

One built-in safety measure of the MAD Merge Tool is that it checks to see if the FULL_STNAME field value is identical for both centerlines being merged. If it is not, the Merge command will abort. It was thought when the tool was designed that most of the time the street names of segments being merged would be the same, and if not, you would have selected the segments in error. This is usually the case, but there are exceptions. In this case, I simply change the street name on the segment that doesn't match to whatever it needs to be, and then perform the merge operation.

After selecting two adjacent centerlines, I press the Merge button on the Centerline MAD Tools toolbar. It doesn't matter if the centerlines are intersected by one or more side streets. After verifying that the street names are identical, I am presented with the window shown on the next page, containing the street names of the centerlines selected.



If you click on one of the entries in the window, the corresponding centerline will flash in the display. I click on the entry whose attributes I want to be retained, and hit OK. The centerline are merged, and the *CMFI* opens (if it wasn't open already). I can then edit the resulting attributes if I need to. At a minimum, the DATE_MOVED value will need to be updated with the current date, as a merge always constitutes a move.

The merge is not committed until the changes are saved.

[Update for second revision: As we have updated our versions of ArcSDE and ArcMap over the last few years, the MAD Merge tool has quit working properly. Therefore, as a workaround, I have to manually delete one of the centerlines, and then modify the geometry and attributes of the other centerline, so that it is the same as if the two previous centerlines had been merged. In the future, we may pay a consultant to update the MAD Split tool, so it will work with the newer versions of the software.]

Cleaning Up the Alley Centerlines

As mentioned on page 102, one of my centerline tasks is to finish cleaning up the alley centerlines. This includes extending them to their adjoining street. In this section, I present two different scenarios that are encountered. In either case, the first thing that must be considered is whether or not the street centerline is aligned with the center of the right-of-way. If not, it will be realigned. This may mean several adjoining street centerlines (and intersections) may need to be moved as well.

A Street And Alley Physically Intersect

The street centerline must be split, which means the resulting street centerlines must be assigned new address ranges. If the alley exists on both sides of the street, but the alley doesn't line up exactly on both sides, the result is an offset intersection. This means the alley centerlines must be warped, so that a logical intersection is created, like the one in Figure 12. A new point must be added to the *Intersection* layer at the logical intersection, and it must be populated. Next, the appropriate centerlines need to be associated to the new intersection. Also, the appropriate centerlines need to be re-associated to the existing intersection that now connects to the newly-created street centerline (that was formed as a result of the split).

Sometimes, as an alternative to an offset intersection, when the alley centerlines on either side of a street are extended to the street, they appear to meet at the exact same spot on the street. However, because I like for things to be exact most of the time, I don't assume that they meet at exactly the same spot. A difference of a few tenths of a foot may not be perceptible unless you zoom in very close. Therefore, since I want the centerlines to follow the exact center of right-of-way as much as possible (even for alleys), but I don't want to have to zoom in very close to every alley intersection to see if the alley centerlines intersect at the exact same point, I have developed a "compromise" procedure for instances such as this.

As an example, consider Figure 118 two pages over. Here Alley 1175 E physically intersects Polk Street from both the north and south. At the scale the map is viewed at, it appears that, for all intents and

purposes, the alley centerlines on either side of the street line up. This is what I do in a situation like this:

First, I extend one of the alley centerlines to the street centerline. It doesn't matter which one, but in this example, I have extended the one south of Polk Street. See Figure 119, next page. The street centerline is split at the intersection, and the attributes (including address ranges) of the resulting street centerlines are revised appropriately. Notice that when the alley centerline is extended, a "pseudo node" is introduced at the original endpoint. This pseudo node is deleted.

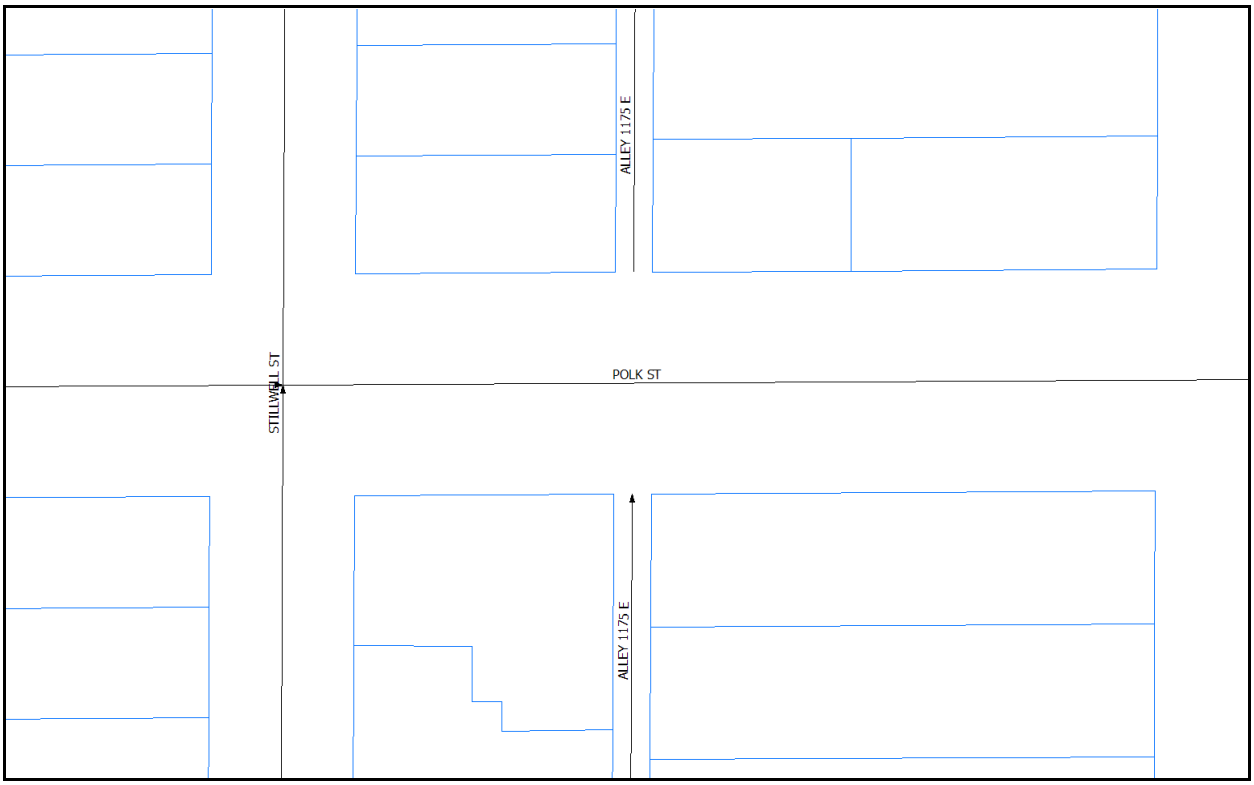


Figure 118 - Example of an alley that appears to line up on both sides of a street.

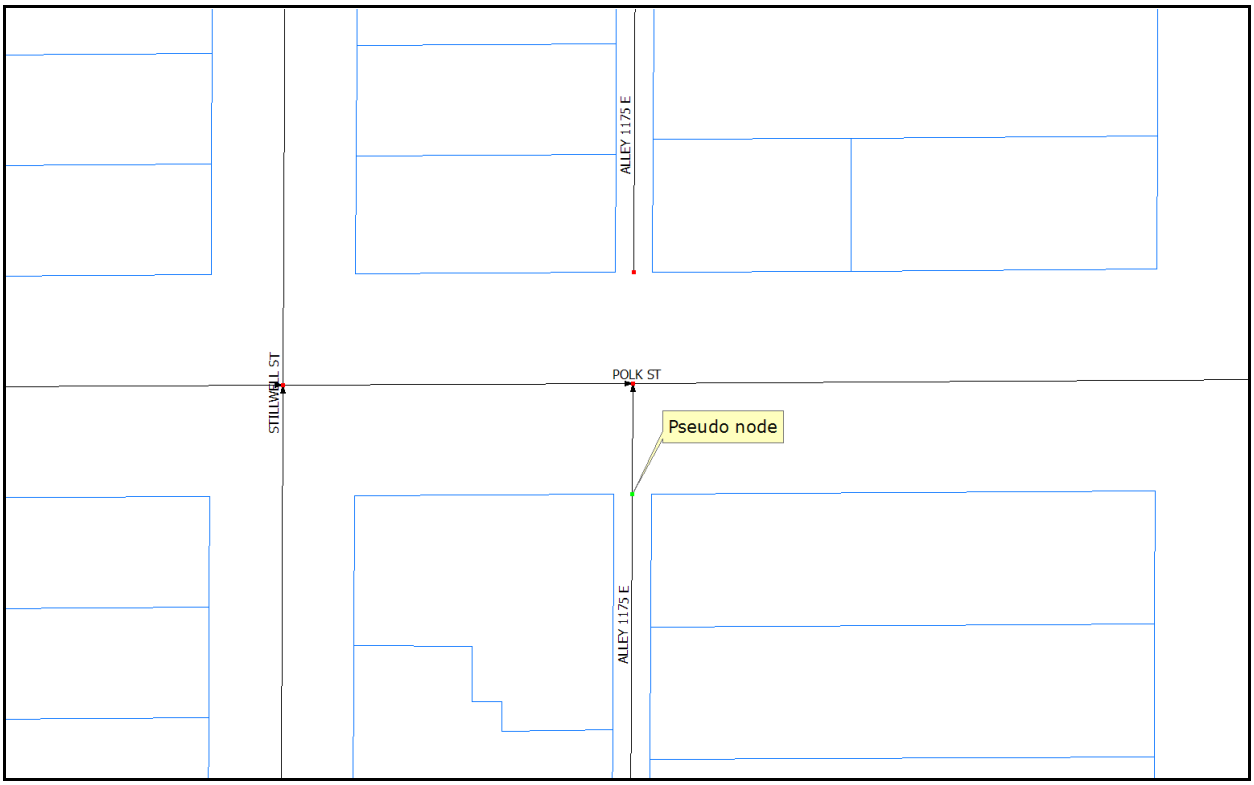


Figure 119 - One of the alley centerlines is extended to the street centerline.

Next I digitize a new centerline from the intersection point to the beginning of the alley centerline on the north side of the street. See Figure 120, next page. Finally, this segment is merged with the existing alley centerline. (Figure 121.) The geometry of the two alley centerlines is now complete.

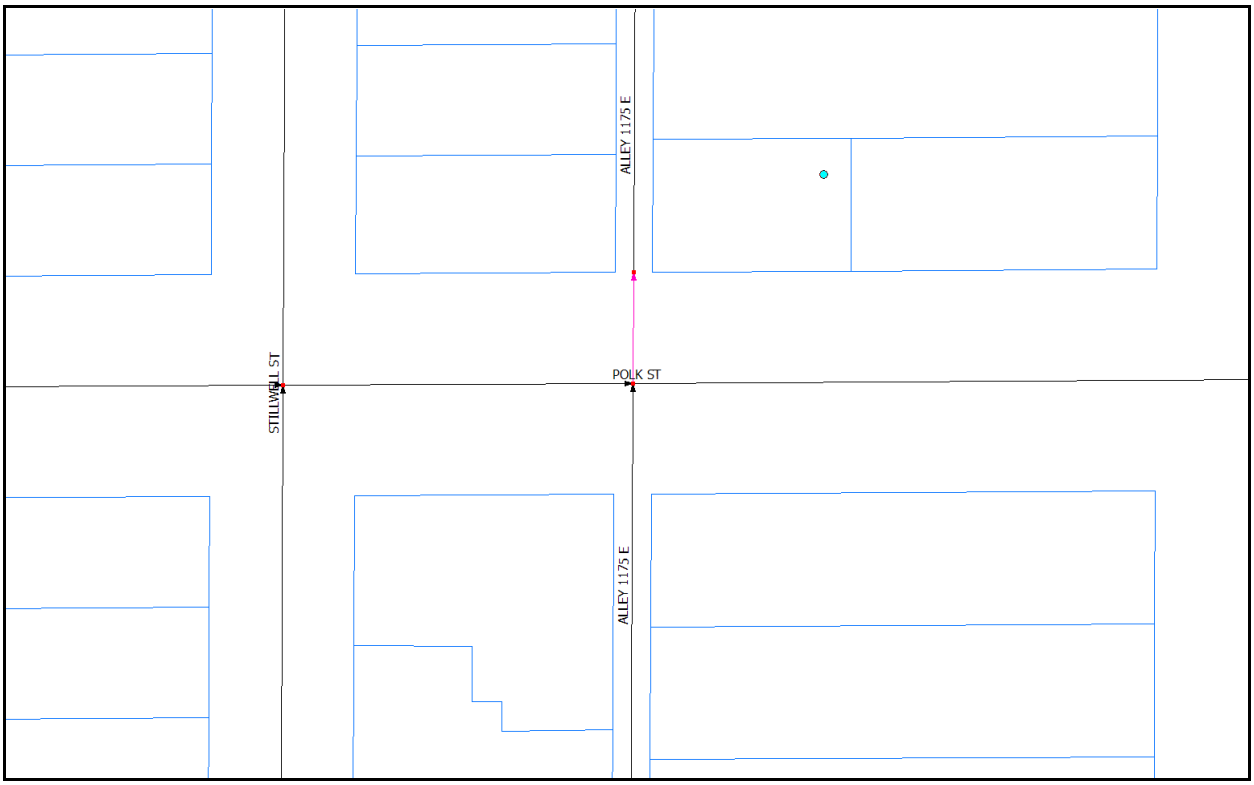


Figure 120 - A new centerline is digitized from the point of intersection to the beginning of the alley centerline on the other side of the street.

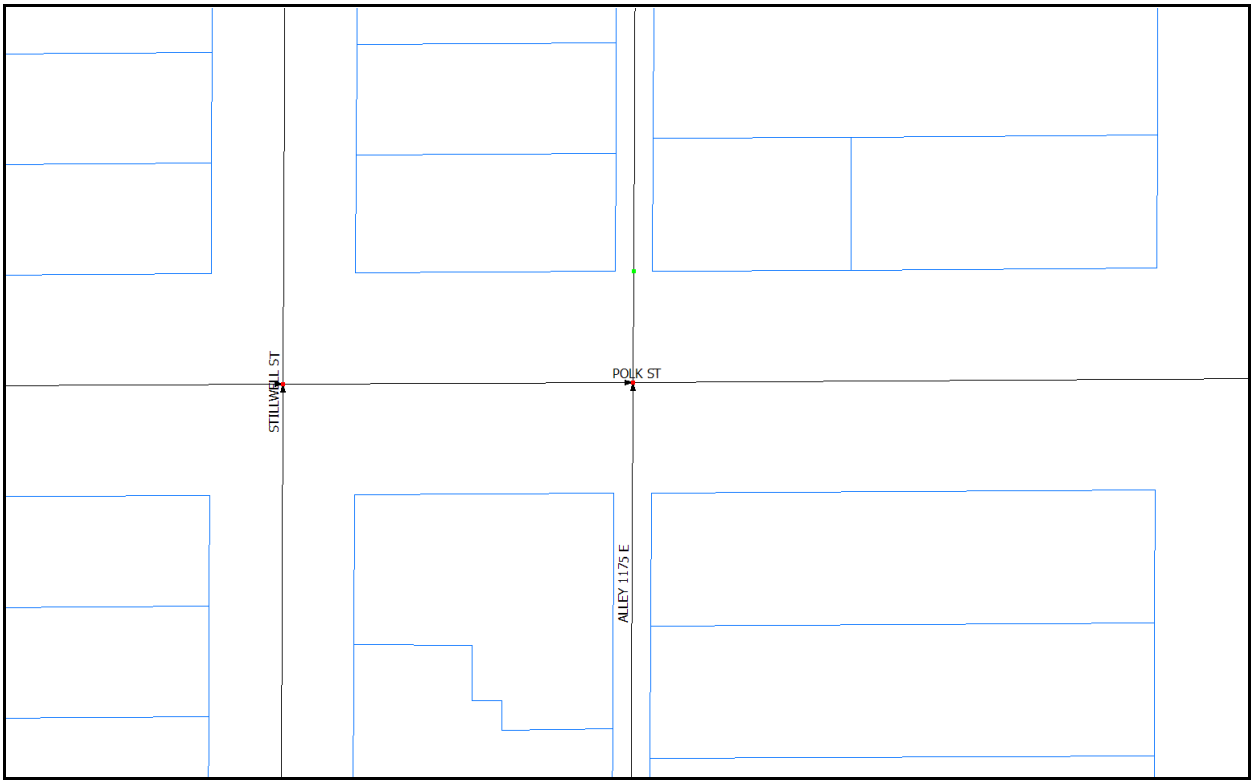


Figure 121 - After merging the alley centerlines on the north side of the street, the alley centerlines have successfully been intersected with the street centerline.

Note in Figure 121 that it appears that a pseudo node is still left on the alley centerline north of Polk Street. However, it is unknown if this is indeed a pseudo node or not, because we do not know if the alley centerlines north and south of the street meet at exactly the same point on the street. They may, or they may be off a foot or less.

However, with the geometry that is the end result of the above procedure, we know two things. First, the alley centerline south of Polk Street follows the center of right-of-way all the way to the centerline of Polk Street. Second, the portion of the alley centerline north of Polk Street that is north of the new vertex follows the center of the right-of-way exactly. In this way, only the portion of the alley centerline between the intersection and the vertex is in question as to whether it follows the center of right-of-way or not. I consider this an acceptable tradeoff to the time involved in extending both alley centerlines, and zooming in very close to see if they line up exactly.

Thus, whenever I come across a situation in the database similar to this (assuming I have the vertices displayed), I know instantly which portions of which centerlines are accurately aligned.

A Street And Alley Don't Physically Intersect

This scenario is similar to the first, except that since there is no physical intersection, the street centerline does not need to be split. It the alley exists on only one side of the street, the alley centerline is simply extended to the street centerline, and the pseudo node resulting from the extension of the alley centerline is removed.

However, if the alley exists on both sides of the street, and neither side is built, the process is more complex. In this scenario, when I say "the alley," I am implying that the name of the alley is the same on both sides of the street, which means that it is the same alley. Now, one of our business rules states that a street (or alley) centerline will not be split where it crosses another centerline, if there is no physical intersection. Therefore, since this is the same alley on both sides of the street, the alley centerlines on both sides need to be merged into one centerline. Consider the example in Figure 122 on the next page.

In this example, Alley 1750 E exists on both sides of Marlowe Avenue, but it is not build on either side. (The operational status has been changed on these centerlines to "PLATTED," so that they will display in the color of magenta, which is indicative of platted centerlines.) Since the alley is platted, this means these centerlines will not be included in most maps, since people are normally not interested in "paper" streets. Therefore, the geometry of the intersection is not as important as if the alley were built. This means that I will dispense with using curves to join the centerlines together (like in Figure 12), but will instead simply use a straight line.

I digitize a straight line between the end of the alley centerline on the south to the beginning of the alley centerline on the north. (Figure 123, next page.) Then I merge this centerline with both the alley centerlines. Usually I pick the alley centerline with the lower address range as the one to be retained after the merge. The result is one alley centerline that crosses the street without intersecting it. Of course, the address ranges of this centerline need to be updated to reflect their new extent. A logical intersection will be placed at the point where the alley centerline crosses the street centerline. (See Figure 124, two pages over.)

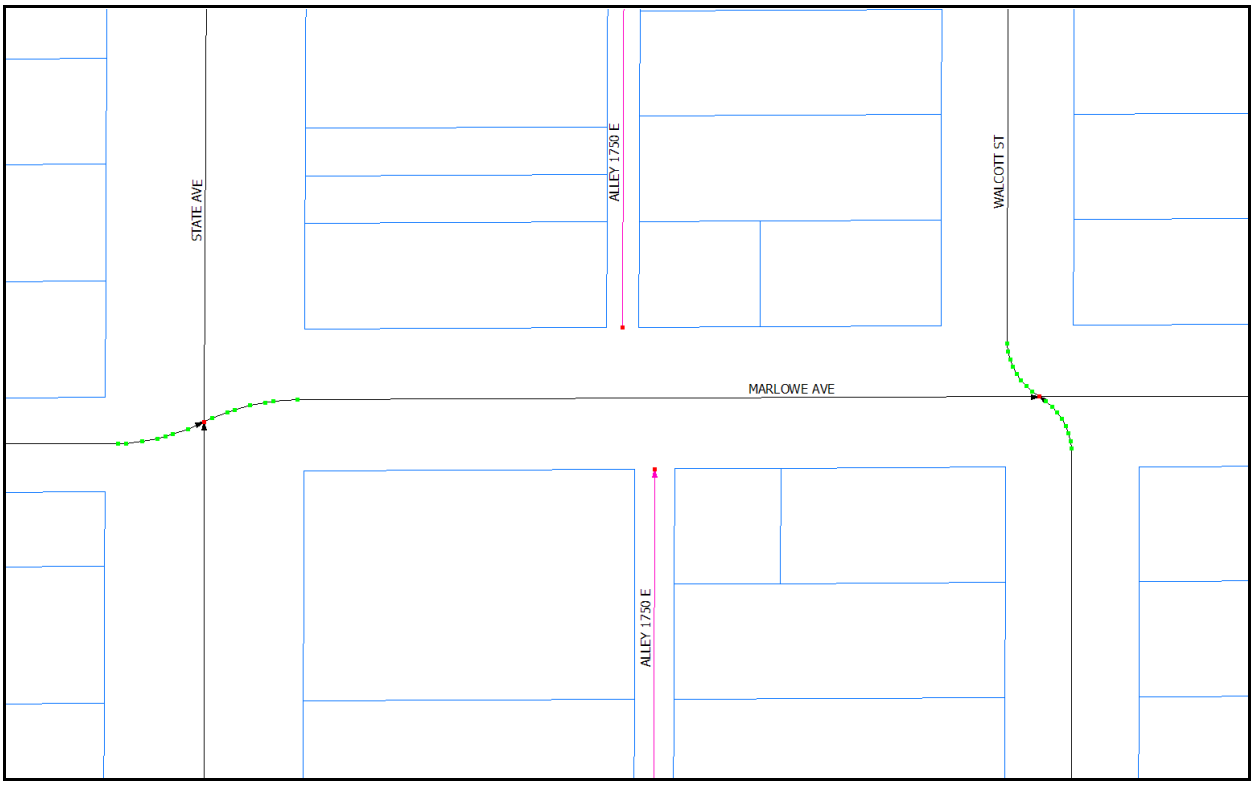


Figure 122 - Two platted alley centerlines that need to be merged together.

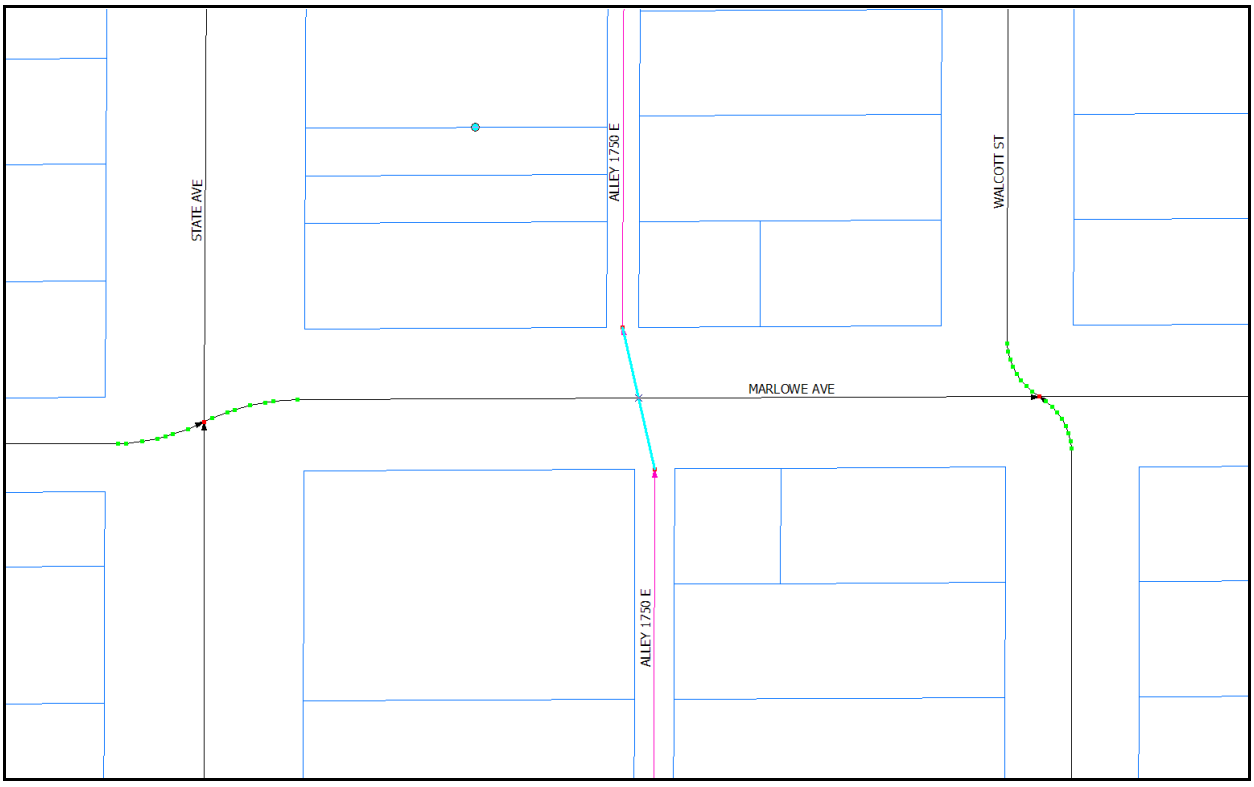


Figure 123 - The alley centerlines are connected with a straight line.

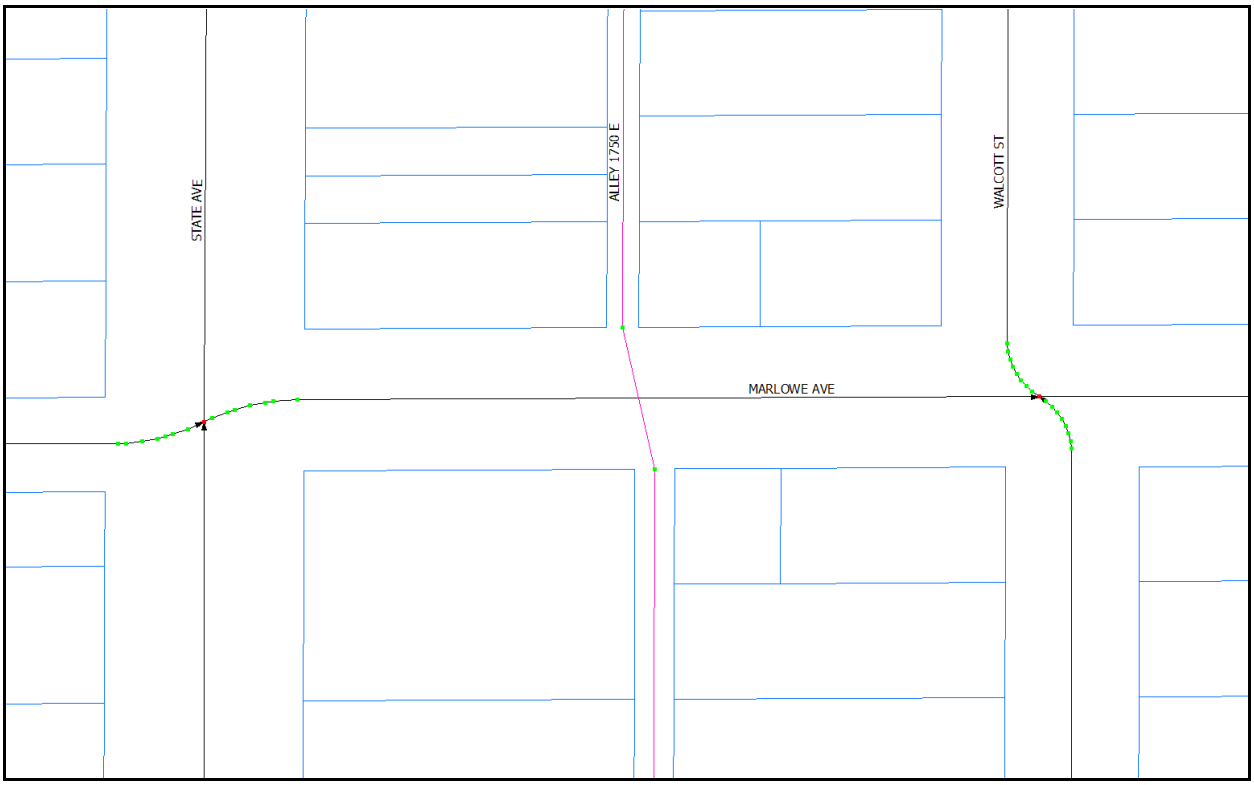
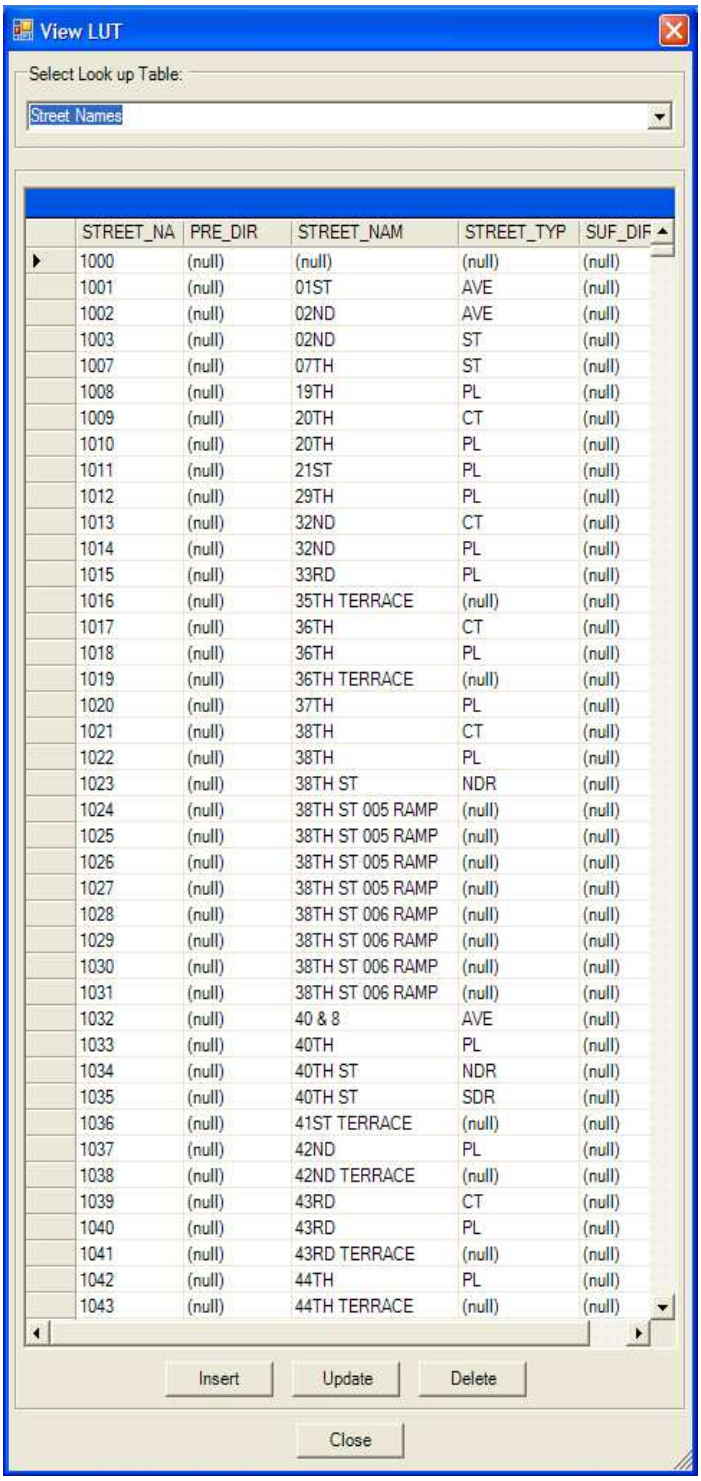


Figure 124 - The alley centerlines have been joined into one.

Adding New Entries to the Lookup Tables

The Street Name Table

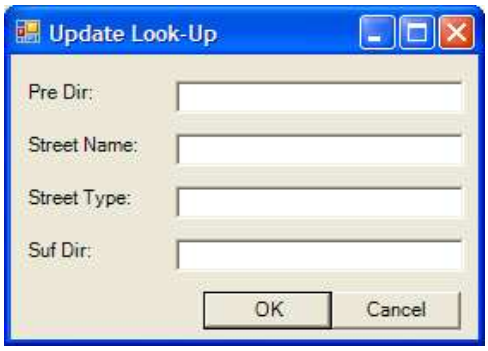
As I mentioned before, before a street name can be assigned to a centerline, the street name must already exist in a table called the Street Name Table. (This includes *all* street names — ramps, alleys, etc.) I can access this table by clicking on the button labeled Open Street Name Table on the MAD Toolbar. It opens in the upper-left corner of my screen on my left monitor. As soon as I open the table, I always click and drag the bottom edge of the table window all the way to the bottom of my screen. This allows me to see the most number of street names possible. I also drag the right edge of the window to the right about an inch. This gives me more room to see the STREET_NAM column in the table. Initially, it is too narrow to see the values in the field in their entirety, so I drag the separator bar between that field and the one to the right (STREET_TYP) till I can read all the street names. The window now looks like Figure 125 on the next page.



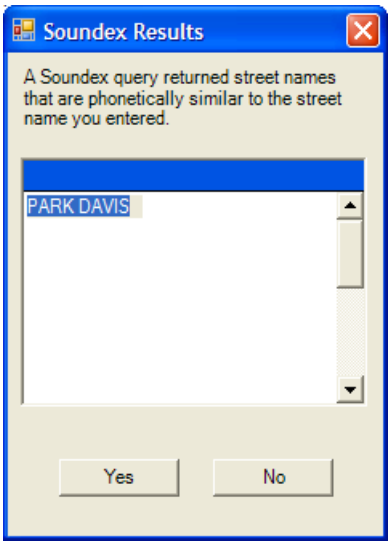
	STREET_NA	PRE_DIR	STREET_NAM	STREET_TYP	SUF_DIR
▶	1000	(null)	(null)	(null)	(null)
	1001	(null)	01ST	AVE	(null)
	1002	(null)	02ND	AVE	(null)
	1003	(null)	02ND	ST	(null)
	1007	(null)	07TH	ST	(null)
	1008	(null)	19TH	PL	(null)
	1009	(null)	20TH	CT	(null)
	1010	(null)	20TH	PL	(null)
	1011	(null)	21ST	PL	(null)
	1012	(null)	29TH	PL	(null)
	1013	(null)	32ND	CT	(null)
	1014	(null)	32ND	PL	(null)
	1015	(null)	33RD	PL	(null)
	1016	(null)	35TH TERRACE	(null)	(null)
	1017	(null)	36TH	CT	(null)
	1018	(null)	36TH	PL	(null)
	1019	(null)	36TH TERRACE	(null)	(null)
	1020	(null)	37TH	PL	(null)
	1021	(null)	38TH	CT	(null)
	1022	(null)	38TH	PL	(null)
	1023	(null)	38TH ST	NDR	(null)
	1024	(null)	38TH ST 005 RAMP	(null)	(null)
	1025	(null)	38TH ST 005 RAMP	(null)	(null)
	1026	(null)	38TH ST 005 RAMP	(null)	(null)
	1027	(null)	38TH ST 005 RAMP	(null)	(null)
	1028	(null)	38TH ST 006 RAMP	(null)	(null)
	1029	(null)	38TH ST 006 RAMP	(null)	(null)
	1030	(null)	38TH ST 006 RAMP	(null)	(null)
	1031	(null)	38TH ST 006 RAMP	(null)	(null)
	1032	(null)	40 & 8	AVE	(null)
	1033	(null)	40TH	PL	(null)
	1034	(null)	40TH ST	NDR	(null)
	1035	(null)	40TH ST	SDR	(null)
	1036	(null)	41ST TERRACE	(null)	(null)
	1037	(null)	42ND	PL	(null)
	1038	(null)	42ND TERRACE	(null)	(null)
	1039	(null)	43RD	CT	(null)
	1040	(null)	43RD	PL	(null)
	1041	(null)	43RD TERRACE	(null)	(null)
	1042	(null)	44TH	PL	(null)
	1043	(null)	44TH TERRACE	(null)	(null)

Figure 125 - The Street Name Table.

The next thing I always do when using this table is click on the STREET_NAM title bar, in order to sort the records by street name. The sort is very fast. Now I can search for a particular street name quickly by dragging the thumb on the right scroll bar up or down. Note that the table contains all four parts that make up a street name, according to the City of Indianapolis' model. Once I determine that the street name I need to add to a centerline is not already in the table, I click the "Insert" button at the bottom of the table window. This brings up the dialog box shown below:



After I fill in the parts of the new street name (being careful to type them in all caps) and click "OK," I am presented with this window:



This window shows the result of a soundex query, in an attempt to find similar street names to the one you typed, to make sure that one of the existing names isn't the one you want. (There is no way to widen the column that shows the results of the soundex, which is a minor flaw.) Since I always check for the existence of the new street name I want, before typing it in, the soundex operation is superfluous as far as I'm concerned, but it acts as a safety check anyway.

Clicking "Yes" inserts the street name I just typed into the Street Name Table. At this point, I can enter more new street names by clicking on the "Insert" button, or click the "Close" button to close the window. New street names must be entered one at a time, which isn't a big deal, because the process is relatively fast, and I never have more than four or five new names to add at one time.

If I need to change a street name (say I misspelled it when I typed it in), I can select it by clicking on the button at the far left side of the street name record in the window, and then clicking the "Update" button at the bottom. This brings up the Update Look-up dialog (the same one that I use to populate new street names) with the street name values filled in. I just change whatever needs to be changed, and click "OK." Changing all or part of a street name in this table does *not* change the name on the centerlines or in the other address records in the MAD. These must be changed separately.

In a similar way, if I want to delete a street name, I select it by clicking its button at the far left, and then click "Delete." Deleting the street name does *not* delete any associated centerlines or address records; in fact, the street name will only be deleted if there are no address records in the MAD associated with that street name. If there are, I will be presented with an error message stating there are child records in the database associated with that street name, and the delete process will be aborted. The idea is not to have orphaned address records in the database (those without a matching street name). To delete a street name with matching address records, all the matching records would have to be deleted first. This very seldom happens. The delete feature here is mainly intended to be used when a new street name is entered incorrectly, before any addresses are assigned to it.

Other Lookup Tables

There are five other lookup tables that are used to validate centerline attributes. They are:

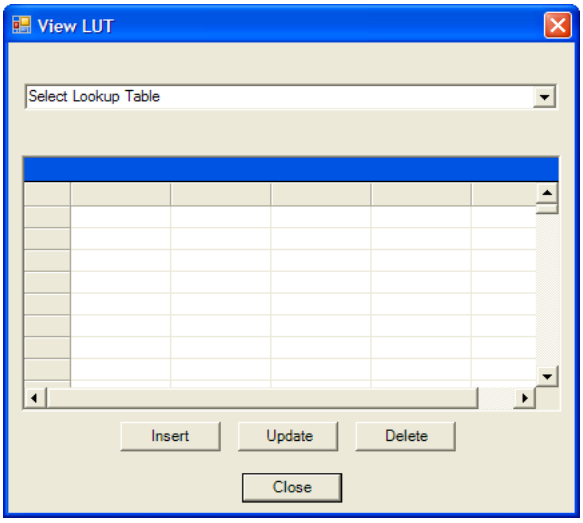
Cities
Counties
Townships
Zipcodes
Addresses

The first four tables contain valid combinations of left and right address-related attributes, for the geographic area of which our Master Address Database encompasses (Marion County, plus a little of Hamilton and Johnson Counties). When the attributes for a centerline are added or revised in the *CMFI*, and the Apply button is pressed, the values in the city, county, township, and zipcode fields are checked against these lookup tables to see if there is a matching combination in each of the tables. For example, suppose I assign a City_Left value of "BEECH GROVE" and a City_Right value of "SPEEDWAY" to a centerline. When I hit Apply, the *CMFI* will return an error message, with these fields highlighted in red, because there is no record in the Cities lookup table with this combination. This is because these cities do not abut each other, so there is no way this combination of values can be valid.

Because these tables list cadastral entities whose boundaries seldom change, they hardly ever need to be modified. Probably the only table whose entities might change (at least in my lifetime) is the zipcode table. I have the ability to add new entries to the lookup tables, if the need arises.

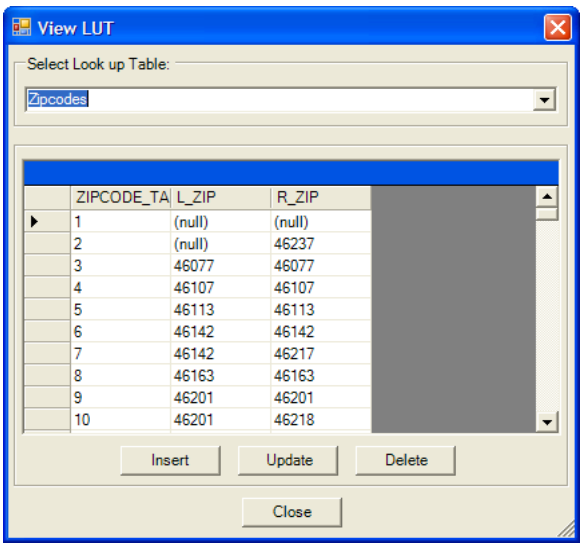
The fifth table contains a list of all valid parcel addresses. I have no need to access this table, so I don't use it.

To access these tables, you click on the Update LUT button on the MAD Toolbar, which brings up the window shown on the next page.



This window also opens in the upper left-hand corner of my monitor. It can be resized, just like the Street Name Table window. Clicking on the picklist gives you the list of tables to choose from that I listed previously. Also included in the list is the Street Name Table that I have already discussed. This is just another way to access that table, but I always access it by clicking on the Open Street Name Table button instead. (Because the Street Name Table is accessed much more frequently than the other tables, it was given its own button to open it.)

Here is what the Zipcode Table looks like:



The buttons in this window functions just like those in the Street Name Table window. I must remember to sort on the appropriate columns, in order to see if a particular combination of values already exists in the table or not, before adding a new record.

Importing New Centerlines

Brian Schneider sends me new centerlines via e-mail, as I discussed in the section on new streets in the *Business Rules* chapter. They arrive as a .dgn file attachment to the e-mail. If the new centerlines will be private, Brian will state this in his e-mail. If he doesn't, I can assume they will be public. After the new .dgn file is renamed, I load the Polyline and Annotation portions of it into my Map Document. I put them in the top of my TOC, immediately below the *Streets* layer. Then I zoom to their extent. I turn on the *Parcel* and *Streets* layers, as well as the Polyline layer for the new centerlines. This way, I can see how the new centerlines fit into the existing fabric of centerline and parcels.

The next thing to do is to select all the Polyline features and copy and paste them into the *Streets* layer. Then I turn off the Polylines; otherwise, I might accidentally select them along with their copy in the *Streets* layer.

Figure 128 on the next page is an example of a new subdivision sent by Brian that is being added. The centerlines to be added from the .dgn Polyline layer are shown in magenta. Also shown is the text that Brian has put into the Annotation layer, in order for me to be able to populate the street names and address ranges. Normally, the parcels for the new subdivision will not have been created by Brian yet; this example was taken from a subdivision that was already put in previously.

After the new centerlines have been copied and pasted into the *Streets* layer, the result looks like Figure 129, next page, with the display of the vertices and endpoints turned on.

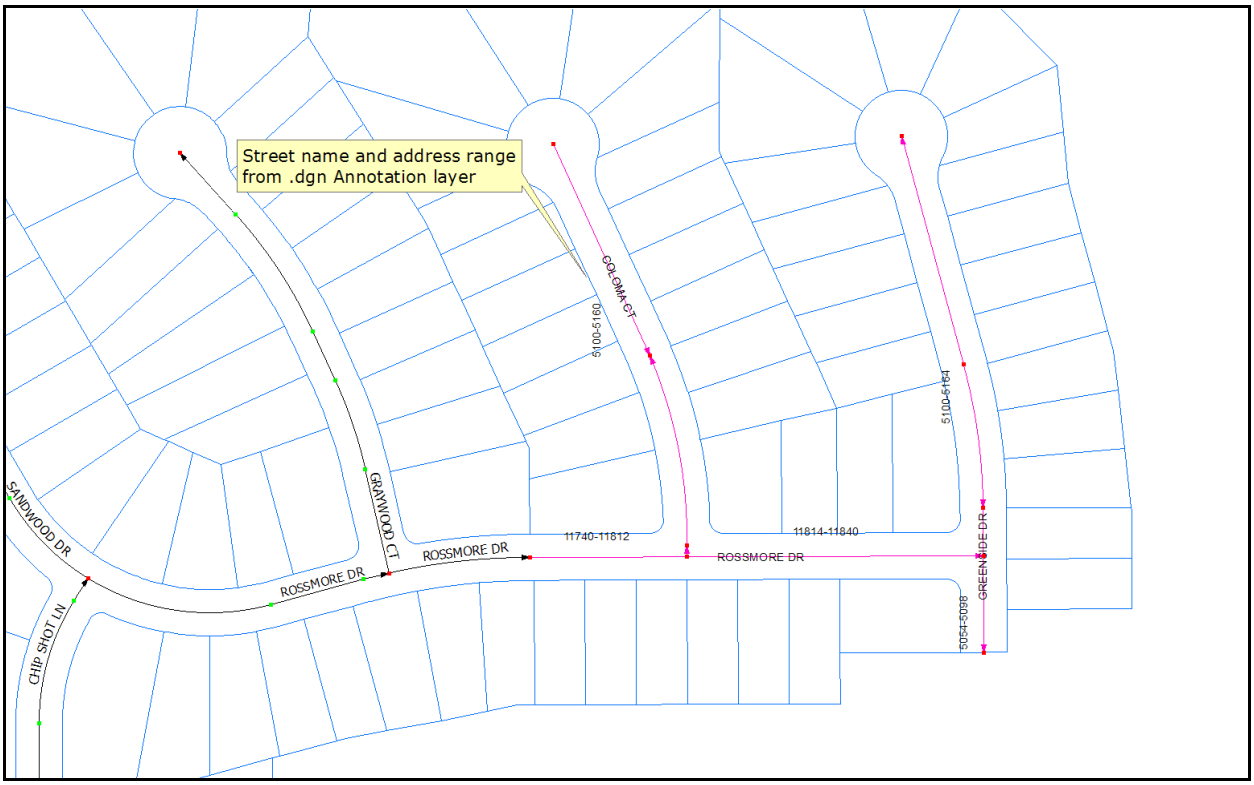


Figure 128 - A new subdivision to be added.

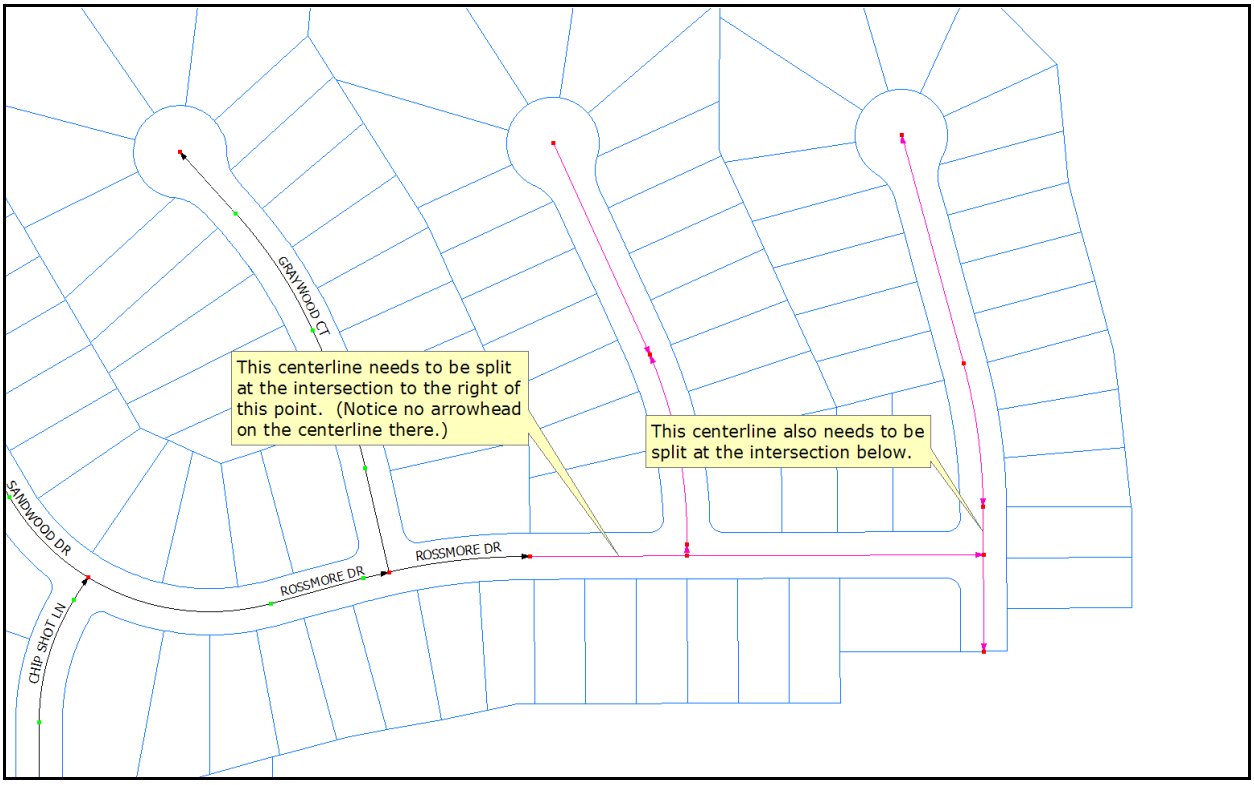


Figure 129 - The centerlines have been pasted into the *Streets* layer from the .dgn file.

In Figure 129, two problems are apparent.

1. Two centerlines are not split where there is a side street intersecting them.

2. The arcs (curves) are separate from the tangent portions. In fact, these two situations are usually the case when I import centerlines from .dgn files.

There are some other common problems associated with centerlines imported from .dgn files, although they are not apparent from Figure 129. They are:

3. Oftentimes the endpoints of the arcs are not snapped exactly to the endpoints of the tangents.

4. Sometimes intersections are not snapped together.

5. Sometimes the entire batch of centerlines don't line up exactly with the existing centerlines that they need to connect to.

So there is always some cleanup to be done to the centerlines before their geometry can be said to be in its final form. I start by taking care of problem one first; that is, splitting the centerlines that need to be split at intersections. (I simply snap to the endpoint of the side street that is intersecting the centerline.)

One thing I need to watch for are streets that change their name when they go around a curve. Usually, when Brian sends me new centerlines like this, the arc will not already be split where the street name changes. Since I don't have anything else to go on, I will go ahead and split the arc at the midpoint. Later, after Brian puts the parcels in with their addresses, I may move the split point, depending on which parcels are addressed off of which street.

Problem #2 is fixed by merging all the appropriate segments together.

In order to fix problems #3 and #4, I need to display the dangles. To do this, I press the Draw Selected Dangles button on the Line Edit Toolbar, the third button from the left. (The centerlines whose dangles you want to see must be selected first.) The result looks like Figure 130 on the next page. The dangles are highlighted by small boxes. In this example, there is an example of problem #3 on Greenside Drive, the street on the far east in the figure. To fix this, I have to zoom in very close on the point where the arc and the tangent are not snapped together (a scale of 1:1 or even tighter!) and snap them.

Anytime a new centerline connects to an existing centerline, the new centerline must be snapped to the old one. This is the situation with Rossmore Drive in the example. Occasionally, the new centerline will be more positionally accurate than the old one, and so the old one must be moved (snapped) to the new one. This will be apparent if there is right-of-way present for the old centerline, but the old centerline is not in the center of the right-of-way. And sometimes there will be some overlap between the old and new centerlines. In this case, one or both of the centerlines must be trimmed, and then snapped together. Again, the layout of the right-of-way will dictate which centerline (or both) needs to be modified.

Sometimes it is apparent that the whole subdivision is off by a small distance (a foot or two). For example, maybe all the new centerlines don't meet existing centerlines they are supposed to tie to, because they are all two feet too far west. I'm not sure why this sometimes happens, but it does. In this case, I will move all the new centerlines exactly the distance they need to be moved in order to line up. I do this with the Move command. Sometimes, in order to find the exact distance that the centerlines need to be moved, I will draw some temporary lines in order to find the exact distance between a centerline intersection and the theoretical point where that intersection should be.

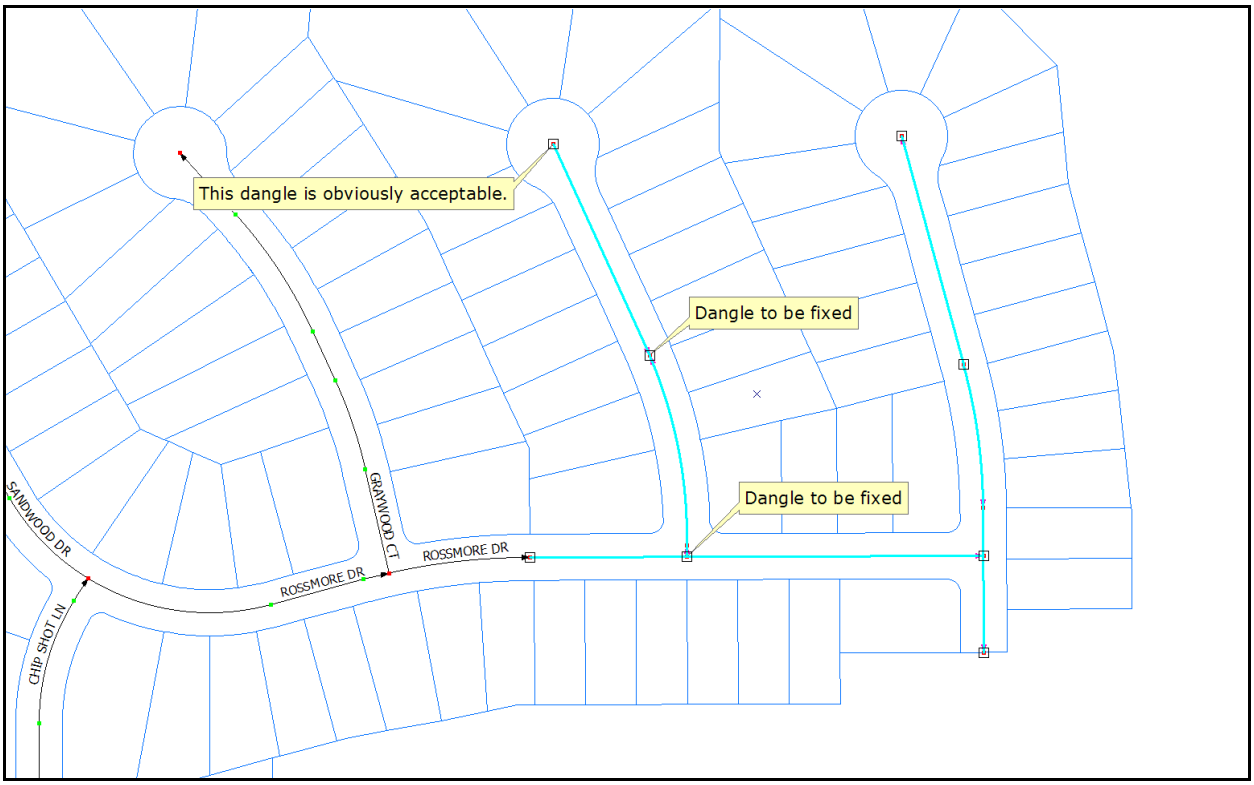


Figure 130 - Checking for overshoots and undershoots.

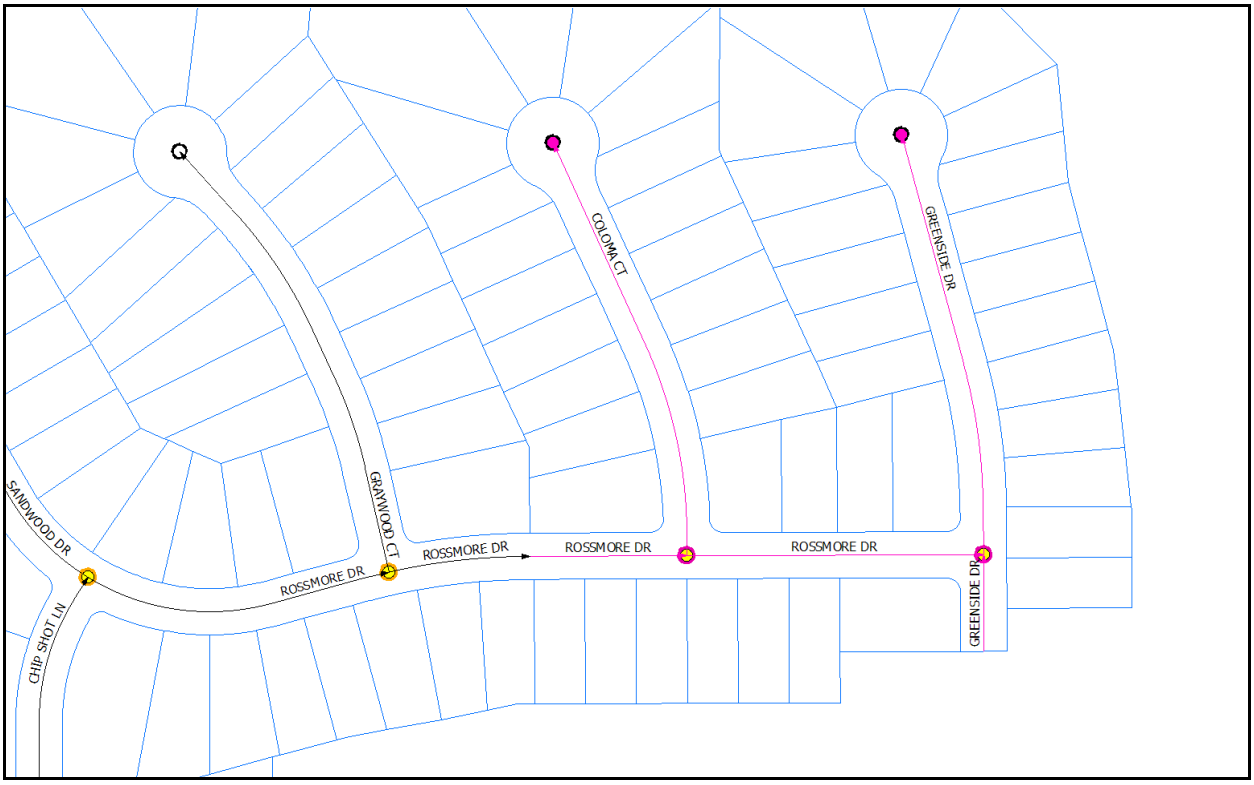


Figure 131 - The finished subdivision.

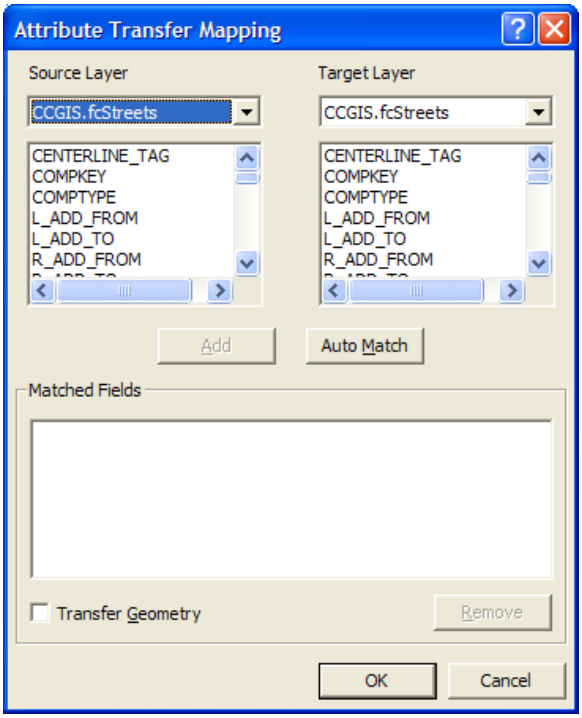
The next thing to do is to split any existing streets that the new centerlines tie into, if there are any. Typically, this happens when new subdivisions tie into thoroughfares. (If a thoroughfare is involved, and the thoroughfare follows a PLSS Section Line, and the thoroughfare centerline doesn't overlay the Section Line, I will move the thoroughfare centerline first, before intersecting it with the new subdivision entrance). Once this is done, the geometry for the new centerlines is finished, assuming all unwanted dangles have been fixed. Then the new centerlines must be examined to ensure they are digitized in the right direction. If not, they will be flipped.

Next, I populate the attributes on the centerlines. I use Brian's annotation to populate the street names and address ranges, and the background layers to populate the rest of the attributes. Once this is done, I send Brian an e-mail informing him that the new streets are in. Then he can begin entering the parcels, with their streets names, addresses, and other attributes. I try to advise Brian the centerlines are ready as soon as I get done with their attributes, because he may be waiting on me to finish before he can proceed.

Next it is time to put the finishing touches on the new centerlines. I add new intersection and cul-de-sac points to their respective layers, if there are any involved, and populate their attributes. Finally, I turn on the *Topology* layer and validate the topology, and save my changes. (I do intermittent saves along the way too.) The finished product resembles Figure 131 on the previous page.

Transferring Attributes

If a centerline (or centerlines) needs to be realigned, sometimes it is easier to redigitize the entire centerline from scratch than to edit it. In this case, I will move the original centerline off to the side, digitize the new centerline, and then transfer the attributes from the original centerline to the new one. This is easily done by clicking the dropdown on the Spatial Adjustment Toolbar and selecting the "Attribute Transfer Mapping..." menu item. This brings up the following dialog:



I click the Auto Match button, which matches all the attributes, because I'm transferring between the same layer. Then I uncheck the Transfer Geometry checkbox, and click "OK." Then I click the Attribute

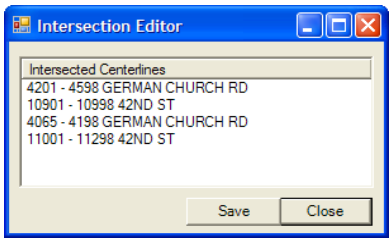
Transfer Tool (the tool at the right end of the Spatial Adjustment Toolbar), click on the centerline I'm transferring the attributes from, and then click the centerline I'm transferring the attributes to. (It helps to have edge snapping turned on, which I normally do.) Using this method, attributes must be transferred one centerline at a time, but the process is quick, and I don't have that many centerlines to contend with at one time. Finally, the old centerline is deleted.

Associating Street Names With Intersections

Recall from the discussion on the *Intersection* layer in the Business Rules chapter that the street names are not attributes of the point features in that layer. This is because of the design of the Master Address Database. Through a series of relationships between the tables in the MAD, it is possible to derive the street names for the intersections. The current thinking is that the *Intersection* points will be exported to a shapefile, and then that shapefile joined to other tables in the MAD in order to attach the street names to the points. It is thought that our users would seldom want to make use of the *Intersection* layer in its native format (SDE).

In order to accomplish the join, there must be an association between the points in the *Intersection* layer and the centerlines for each intersection. Again recall from the Business Rules chapter that the intersections are "logical" intersections; they do not always lie at the physical intersection of centerlines. There is no way to automatically determine which centerlines participate in a logical intersection, so these associations must be created manually. This is the purpose of the MAD "Intersection Editor" tool, the third button from the right on the MAD Toolbar (page 110).

To use this tool, you first select an intersection and then press the tool button. (If you don't select an intersection first, or select more than one intersection, you will get an error message to that effect.) A dialog box will appear with a list of all the centerlines that physically intersect at that intersection.



All the centerlines that are listed in the dialog box will also be selected in the Map Document, so you can easily tell which ones are involved in the intersection. The intersection itself will also remain selected. If the list of centerlines for the selected intersection is correct, which in 95% of the cases will be, you click Save. It is not necessary to close the dialog box before selecting another intersection, but you must click Save after each intersection. Clicking Save creates the relationship in the MAD database between the intersection INTERSECTION_TAG attribute and the associated centerlines. It does *not* save your work; you still need to save your edits in order for the relationships to be stored permanently.

After you select an intersection, if you need to add one or more centerlines to the intersection that did not physically intersect at the intersection point, you simply hold down the <Shift> key and click and drag with the ESRI Edit Tool to select the centerlines you want to add to the intersection (the selected set). This is the same operation you normally employ to add features to your selection set. Conversely, if you want to remove one or more centerlines from the intersection, you simply click and drag across the centerlines you want to remove.

Viewing Parcels Associated With Centerlines

The MAD Toolbar contains a button labeled "Show Parcels Related to Centerline." If a centerline is selected, pressing this button will highlight all the parcels whose street name matches the selected centerline, and whose address falls within one of the address ranges on the centerline. Only one centerline at a time may be selected. I use this button infrequently.

Deleting Centerlines and Intersections

This section refers to deleting the entries in the MAD that are associated with centerlines and intersections, as well as the geographic features themselves. If only the geographic features need to be deleted, then the generic ESRI tools for these operations are sufficient. However, the only time this situation is normally present is when the features to be deleted are temporary, or they have been created by mistake. Even then, they should only be deleted with the ESRI tools if no MAD records have been stored with them; that is, using the *CMFI*.

If there are MAD records associated with the centerlines or intersections, then the MAD Delete Centerline and MAD Delete Intersection tools on the Centerline MAD Toolbar must be used. These tools will delete whatever feature is selected, along with its associated records in the MAD. Only one feature at a time can be deleted. However, a centerline will not be deleted if it has any child records in the MAD associated with it. These are records containing addresses of other features (such as parcels) that fall on the centerline. In this case, a message will appear informing me there are child records present. If I decide that a centerline has been entered in error, and addresses have also been assigned to it in error, then I would have to enlist the help of others with edit rights to the MAD database to delete the child records before the centerline could be deleted.

GOALS

There are many goals for the centerlines, in order to improve them and make them more useful. Some of the goals are already underway. The tables on the following pages attempt to summarize the current goals.

High Priority Goals

Goal	Priority	Resources Used	Status
Provide centerlines for all improved streets in Marion County (or at least one centerline for streets with branches, such as in apartments), as well as those streets that are planned in the near future, or currently under construction	High	Aerial photography, pavement layer, parcel layer, new centerlines from Brian Schneider, *Official Thoroughfare Plan*, other	Complete and ongoing
Enter new centerlines as they are provided to me	High	Brian Schneider	Complete and ongoing
Enter new vacations, acceptances, and speed limit, one-way, and weight limit ordinances as they are sent to me	High	Vacation, acceptance documents, ordinances	Complete and ongoing
Change operational status of "platted" streets to "built" when improved	High	Aerial photos, others?	Complete and ongoing

Medium Priority Goals

Goal	Priority	Resources Used	Status
Eliminate nodes at intersections of built and non-built streets	Medium	Aerial photos	Approximately 85% complete. I fix them as I find them.
Move all centerlines where right-of-way is present to follow the center of the right-of-way or center of pavement (whichever the case may be), unless extenuating circumstances are present	Medium	Parcel layer	Ongoing. I fix these in the course of my other duties.
Add centerlines for all alleys	Medium	Parcel layer, aerial photos	Preliminary alley centerlines are done. Final cleanup approximately 30% complete.
Finish populating the L_TRACT and R_TRACT attributes	Medium	Census00 layer	Approximately 90% complete.
Complete and/or verify address ranges for apartments and condominiums.	Medium	*Units* layer	Ongoing. Pike Township finished.
Populate zipcode and township attributes for out-of-county centerlines.	Medium	*Nine-county zipcodes*, *Nine-county townships* layers.	Ongoing. Approximately 50% complete.

Low Priority Goals

Goal	Priority	Resources Used	Status
Change "left-to" or "right-to" address ranges for streets that end in exact hundred blocks to the even hundred block number	Low	Centerlines	Less than 1% complete
Create a script that can be run against the centerlines to check for various types of logical errors in the attributes	Low	Centerlines	Already started, but I haven't worked on this in a long time. I used to have an AML that did the same thing.
Research "unknown" street names	Low	Historical maps, Assessors' plats, others?	A few have been done.
Enter street acceptance dates of old streets	Low	Lists of street acceptances I obtained from Dave Gillman of DMD (now in my office) and Brian Schneider	Not yet started
Enter old street names on streets that have changed names	Low	Lists of street name changes available from Brian Schneider	Not yet started
Enter missing centerlines of vacated streets, along with Vacation Petition number	Low	DMD files, DPW databases	Not yet started
Reconcile centerlines with Pavement Management System	Low	Pavement Management System (PMS)	Preliminary Block_ID's have been populated on the centerlines, but there is currently no process in place to maintain them, as centerlines are split or merged. Occasionally I receive updates from Sherry Powell, the custodian of the PMS.
Create a turntable for routing purposes.	Low	*Consolidated Code of the City of Indianapolis*	Not yet started